Adv Mobile Learn Educ Res, 2026, 6(2): 1857-1874
DOI: 10.25082/AMLER.2026.02.003

CASE STUDY

Mobile Learning for Programming Education: A Case Study of SoloLearn and Self-Directed Learning Skills

Daniel Danso Essel

Department of ICT Education, University of Education, Winneba, Ghana

Correspondence to: Daniel Danso Essel, Department of ICT Education, University of Education, Winneba, Ghana; Email: ddessel@uew.edu.gh

Received: November 14, 2025;
Accepted: July 4, 2026;
Published: July 8, 2026.

Citation: Essel, D. D. (2026). Mobile Learning for Programming Education: A Case Study of SoloLearn and Self-Directed Learning Skills. *Advances in Mobile Learning Educational Research*, 6(2), 1857-1874. https://doi.org/10.25082/AMLER.2026.02.003

Abstract: This study investigates how mobile learning supports self-directed learning (SDL) in programming education through a case study of SoloLearn. A cross-sectional survey of 708 undergraduate students at the University of Education, Winneba examined app usage, perceptions, and programming confidence, framed by Knowles' (1975) SDL Theory and Fredricks et al.'s (2004) Student Engagement Framework. Partial least squares structural equation modeling (PLS-SEM) revealed a sequential pathway from self-management to motivation, monitoring, strategy use, and academic performance. Most participants (46.8%) reported increased confidence, while 38.4% reported improvements in real-world application skills, though challenges included limited advanced content, ad disruptions, and insufficient feedback. The findings suggest that SoloLearn effectively develops foundational SDL skills but requires adaptive features, project-based modules, and improved collaborative tools to support deeper learning. Results should be interpreted cautiously due to the single-institution, male-dominated sample. Overall, the study makes three contributions: it provides empirical evidence of a sequential SDL pathway from self-management to academic performance in mobile programming education; it demonstrates the value of combining objective grade data with self-report measures in mobile learning research; and it offers practical guidance for integrating mobile coding platforms into programming instruction across diverse educational contexts.

Keywords: mobile learning, self-directed learning, programming education, SoloLearn, student engagement, mobile coding apps, self-regulated learning

1 Introduction

The rapid digital transformation of education has accelerated the global adoption of mobile learning (m-learning) as a key enabler of accessible, flexible, and personalized instruction. As mobile platforms increasingly incorporate features designed to support personalized and self-paced learning, they are transforming how learners engage with content across diverse educational settings from rural Ghana to urban classrooms in the Global North. Although some platforms are beginning to integrate adaptive features, the present study focuses specifically on students' perceived experiences of using SoloLearn as a mobile programming learning environment, rather than on any AI-driven features of the platform.

In programming education, where iterative practice, problem solving, and mastery of abstract concepts are central, mobile platforms offer unique opportunities to extend learning beyond traditional classrooms. Among these, SoloLearn has emerged as a globally adopted mobile coding application offering tutorials, assessments, and peer interaction within a gamified, self-paced environment. Its alignment with self-directed learning (SDL) principles such as goal setting, autonomous exploration, and iterative self-assessment makes it particularly relevant to programming, where learners must independently debug, test, and refine solutions.

SDL refers to a process in which learners take active responsibility for diagnosing their needs, setting goals, choosing resources, and evaluating progress (Knowles, 1975). This competency is especially critical in computing education, where persistence and autonomy underpin effective problem solving. While prior studies affirm that mobile learning can support SDL by enabling engagement and flexible practice (Dahri et al., 2023; Shin & Kang, 2015), most focus on general education or language learning contexts. Few address computing-specific SDL development, and even fewer examine how AI-powered mobile platforms like SoloLearn contribute to skill acquisition, motivation, and learner autonomy across varied institutions and regions.

Given the growing importance of scalable, mobile-first programming education particularly in low-resource environments and cross-institutional curricula, this study provides timely evidence on how mobile platforms support core SDL processes. By examining SoloLearn through the lens of Ghanaian undergraduates and situating the findings within the global discourse on mobile computing education, the study offers practical insights that are transferable across cultural and institutional contexts.

This research investigates how SoloLearn supports the development of SDL competencies among undergraduate programming students. Specifically, it explores engagement patterns, perceived usability and effectiveness, and the link between app use and students' confidence in independent problem solving. Drawing on the SDL theory of Knowles (1975) and the Student Engagement Framework of Fredricks et al. (2004), the study integrates cognitive and motivational perspectives to understand how mobile learning environments foster autonomy, confidence, and sustained learning behaviors in programming education.

1.1 Research Objectives

(1) To examine how the SoloLearn mobile app supports the development of self-directed learning (SDL) skills among undergraduate programming students.

(2) To explore the relationship between students' use of SoloLearn features and their confidence in solving programming tasks independently.

(3) To assess students' perceptions of the usability, effectiveness, and educational value of SoloLearn in programming education.

(4) To identify challenges and opportunities associated with integrating mobile learning apps into programming instruction.

1.2 Research Questions

(1) How does the use of SoloLearn contribute to the development of self-directed learning skills in programming?

(2) What are students' perceptions of SoloLearn's usability, interactivity, and effectiveness for learning programming?

(3) To what extent does SoloLearn use influence students' confidence and ability to solve programming problems independently?

(4) What challenges do students face when using SoloLearn for programming education, and how can the app be improved to better support SDL?

1.3 Problem Statement

Despite the growing adoption of m-learning applications in higher education, many students continue to face challenges in mastering core programming skills particularly the SDL competencies essential for success in this field. Programming requires sustained problem-solving, independent debugging, and the application of abstract concepts, all of which demand active, self-regulated learning rather than passive content consumption.

Mobile platforms like SoloLearn offer flexible, interactive, and self-paced environments that appear well-suited to support SDL. However, empirical evidence on their effectiveness in fostering discipline-specific SDL skills in programming remains limited. Existing research has largely focused on general benefits of m-learning or user satisfaction, often overlooking how such tools influence autonomous learning behaviours specific to programming education.

The existing gap prevents educators and curriculum designers from effectively planning mobile tool integration into programming instruction. The lack of understanding about how SoloLearn supports SDL development prevents educators from maximizing its potential for independent learning and potentially leads to misalignment with educational goals.

This research examines SoloLearn's contribution to undergraduate students' self-directed programming abilities to develop practical recommendations for improving mobile learning approaches in computer science education.

1.4 Significance of the Study

This study contributes to programming education, mobile learning research, and instructional design by examining how a widely used mobile coding platform can support the development of self-directed learning (SDL) skills. It provides empirical insights into how features such as gamification, feedback, and flexible access may influence learner motivation, engagement, and independent problem-solving in programming contexts.

The study offers practical implications for educators and curriculum designers by highlighting ways mobile learning tools can be integrated into programming instruction to support learner autonomy and skill development. It also contributes to educational technology research by examining how mobile learning environments may be designed to better support SDL processes. By focusing on programming education within a mobile learning context, the study addresses a gap in existing literature and provides insights that may inform both research and practice in diverse educational settings.

1.5 Theoretical Framework

The study is grounded in Self-Directed Learning Theory of Knowles (1975) and Student Engagement Framework of Fredricks et al. (2004) . Together, these frameworks explain how mobile learning platforms can foster both autonomy and sustained engagement, which are essential for programming education.

1.6 Self-Directed Learning Theory

Knowles (1975) conceptualized SDL as a cyclical process involving diagnosing needs, setting goals, selecting strategies, and evaluating outcomes. Programming practice, with its emphasis on iterative problem-solving and debugging, closely aligns with this process. SoloLearn supports SDL by enabling learners to progress at their own pace, set personal goals, and receive real-time feedback, thereby reinforcing persistence and autonomy.

1.7 Student Engagement Framework

Fredricks et al. (2004) identify behavioral, emotional, and cognitive dimensions of engagement. In programming contexts, these dimensions can be observed in activity persistence, enjoyment of gamified features, and reflective problem-solving. Applying this framework helps clarify how mobile platforms sustain learner motivation and promote deeper processing of coding concepts.

1.8 Integrating SDL and Engagement Frameworks

Combining SDL with engagement theory provides a comprehensive lens for analyzing SoloLearn. SDL highlights the developmental process of independent learning, while engagement theory explains the behaviours and affective states that sustain this process. This integration allows for examining both learner agency and design features (e.g., feedback, gamification) in shaping programming skill development.

2 Literature Review

Mobile learning has become an established mode of technology-enhanced education, valued for its flexibility, accessibility, and potential to support personalized learning. Research shows that m-learning environments can foster self-directed learning (SDL) by enhancing learner autonomy, motivation, and engagement (Deng & Gao, 2023; Ifenthaler & Schumacher, 2016). SDL, defined as the capacity to plan, implement, and evaluate one's learning (Knowles, 1975), has been associated with improved persistence and performance in online and blended contexts. However, much of this evidence is derived from general education or language learning, offering limited insight into programming education, where learners face unique challenges such as abstraction, debugging, and sustained problem-solving. Recent evidence suggests that many mobile coding applications still rely on static content, while more interactive features such as coding environments, quizzes, and feedback mechanisms are more effective in supporting engagement and learning outcomes (Schnieder & Williams, 2023). However, there remains limited empirical evidence on how such mobile learning features support the full process of self-directed learning in programming contexts. This highlights the need to examine how SoloLearn supports SDL processes and learning outcomes among undergraduate programming students.

Programming education presents unique cognitive and motivational barriers. Students often struggle to maintain engagement due to the conceptual complexity of programming syntax and the iterative nature of debugging, which can lead to frustration and high dropout rates (Salleh et al., 2013). In response, mobile coding applications have emerged as accessible platforms that combine gamification, microlearning, and community interaction to support continuous practice and self-regulation. Oyelere et al. (2017) demonstrated that mobile coding tools significantly improved learner motivation and achievement within African higher education, showing the potential of mobile-based instruction for low-resource contexts. Similarly, Calderon et al. (2023) and Caldern-Garrido et al. (2022) emphasized the importance of active learning strategies such as flipped instruction and project-based tasks supported by mobile technologies to enhance student engagement and skill mastery. Amro and Romli (2019) further compared popular coding apps, including SoloLearn and Programming Hub, revealing that gamified designs and feedback mechanisms play a decisive role in sustaining motivation and persistence.

Recent app-based programming research suggests that the educational value of mobile coding platforms depends less on content availability alone and more on the quality of interaction they provide. Schnieder and Williams (2023) review of 78 Python-learning mobile apps found that over one-third of apps have static content, resembling digital textbooks, while apps with multiple dynamic features received higher user ratings. Features such as quizzes, IDEs, certificates, community support, and competitions were identified as important design elements. In particular, apps with built-in IDEs were downloaded significantly more often, suggesting that learners value opportunities for direct coding practice. These findings reinforce the need to evaluate platforms such as SoloLearn not only as mobile content repositories but as interactive environments that can support practice, feedback, motivation, and self-directed learning.

Recent research is increasingly focusing on how mobile learning environments can integrate adaptive, AI-powered, and analytics-driven features to promote deeper self-regulated learning. Roll et al. (2021) argue that AI-driven scaffolding enables dynamic feedback loops that adapt to learners' cognitive states in real time, improving engagement and performance. Giannakos et al. (2022) highlight the growing use of multimodal learning analytics in mobile education to capture behavioral and emotional engagement patterns. Similarly, Ifenthaler (2021) and Broadbent and Poon (2015) show that learning analytics and self-regulation strategies predict long-term achievement in digital learning ecosystems. Within computing education, Zheng et al. (2026) found that project-based mobile learning approaches enhance learners' ability to transfer coding skills to real-world tasks, while Spirina (2024) reported that interactive mobile platforms foster readiness for professional software development roles. These findings suggest that mobile coding tools are evolving beyond static content delivery toward intelligent, data-informed ecosystems capable of personalizing learning trajectories.

Gamification research provides further insight into how mobile platforms can sustain motivation in self-directed contexts. Sailer and Homner (2019) and Ryan and Deci (2017) note that features such as points, badges, and leaderboards can strengthen intrinsic motivation when aligned with learners' needs for autonomy and competence. In the context of programming education, such gamified mechanisms facilitate iterative practice, feedback seeking, and persistence; behaviours closely tied to SDL. Nonetheless, not all learners benefit equally as Ifenthaler and Yau (2020) caution that gamified or competitive environments may disadvantage learners with lower prior confidence, emphasizing the need for adaptive scaffolding that balances challenge with support.

Despite these advances, the intersection of mobile learning, AI-based personalization, and SDL development remains underexplored in programming education. Most existing studies emphasize usability or motivational outcomes without modeling the underlying relationships among engagement, metacognition, and academic performance. Moreover, evidence is disproportionately concentrated in Western and Asian contexts, leaving limited understanding of how mobile platforms function in African higher education systems where infrastructure and learner needs differ. Addressing this gap, the present study applies Partial Least Squares Structural Equation Modeling (PLS-SEM) to examine how SoloLearn supports SDL competencies among undergraduate programmers. By combining quantitative and qualitative data, it provides a holistic understanding of how app-based learning behaviors relate to motivation, self-monitoring, and perceived academic achievement. The findings aim to generate context-sensitive yet globally applicable insights for computing educators, curriculum designers, and mobile learning developers working toward adaptive, scalable programming education models.

3 Methodology

3.1 Research Design

This study adopted a quantitative cross-sectional survey design complemented by qualitative open-ended responses to examine how the SoloLearn app supports the development of self-directed learning (SDL) skills in programming education. The design was appropriate for exploring relationships among learners' motivation, engagement, and perceived learning outcomes at a single point in time. While this design allows for broad data collection from a large cohort, it reflects students' perceptions rather than direct measurements of programming competence, a limitation acknowledged in the interpretation of findings.

3.2 Population and Sampling

The target population comprised two consecutive year groups of Level 200 students in the Department of ICT Education at the University of Education, Winneba, Ghana, who had completed Object-Oriented Programming (OOP) courses using Java. A convenience sampling approach was adopted. Google Forms survey links were made available to students during class sessions across two consecutive academic years, and participation was voluntary. From a total of 910 students across both cohorts, 708 valid responses were obtained, yielding a response rate of 77.8%. This high participation supports the internal reliability and contextual validity of the study, although results may not be generalizable beyond the institutional and cultural context. Although all questionnaire items were set as required in Google Forms to minimise incomplete submissions, a total of 202 responses were excluded from the final dataset due to incompleteness, wrong or duplicate submissions identified during data screening. The remaining 708 valid responses were retained in full for analysis, with no further exclusions based on statistical outlier detection.

3.3 Instruments

A structured questionnaire was designed to capture key constructs derived from the Technology Acceptance Model (TAM), Self-Efficacy Theory, and the Student Engagement Framework. While SDL theory and the Student Engagement Framework served as the primary theoretical lens guiding the study, TAM and Self-Efficacy Theory specifically informed the design of questionnaire items related to perceived ease of use, usefulness, and programming confidence. This distinction clarifies the complementary role of these frameworks within the overall study design. SoloLearn was integrated into the course structure as a required prerequisite and was subsequently used throughout the semester to support micro-learning activities and formative assessments. This ensured that all participants had structured and meaningful exposure to the platform prior to completing the survey. The instrument consisted of closed-ended items on app usage, perceived ease of use, usefulness, motivation, self-efficacy, and academic outcomes, each rated on a five-point Likert scale (1 = Strongly Disagree to 5 = Strongly Agree). Three open-ended items were included to gather qualitative data on students' perceived strengths, weaknesses, and improvement suggestions for SoloLearn. Expert validation by three ICT education specialists and a pilot test with 35 participants ensured the questionnaire's clarity and content validity (Cronbach's $\alpha = 0.88$).

3.4 Data Collection Procedure

Data were collected via a secure Google Forms survey distributed during the revision week of the semester. Participation was voluntary, anonymous, and ethically approved by the University's Institutional Review Board. Informed consent was obtained digitally, and confidentiality was assured. The survey was administered over two weeks, capturing both quantitative data and qualitative reflections from participants' experiences using SoloLearn.

3.5 Data Analysis

Data analysis was conducted using Partial Least Squares Structural Equation Modeling (PLS-SEM). This approach was selected because the study focuses on predicting relationships among key constructs and exploring extensions of existing theoretical frameworks. In this study, relationships among motivation, self-monitoring, engagement, and perceived academic performance are examined within an SDL framework, making PLS-SEM an appropriate analytical choice. Originally introduced by Wold (1982) and further developed by scholars such as Hair et al. (2021), PLS-SEM is a variance-based technique designed for prediction-oriented research, particularly in situations where data may not fully meet strict statistical assumptions. Unlike covariance-based SEM (CB-SEM), which typically requires large samples and normally distributed data, PLS-SEM is more flexible and aims to maximize the explained variance of dependent constructs (Hair et al., 2019). In addition, PLS-SEM is well suited for analysing complex multi-construct models involving multiple relationships (Akter et al., 2017). Given that this study investigates a multi-construct model and seeks to extend understanding of SDL processes in mobile programming contexts, PLS-SEM provides a robust and appropriate method for modelling these relationships (Hair et al., 2019).

Each SDL construct was operationalised using reflective indicators drawn from the questionnaire. Self Management was measured by frequency of app use and weekly hours of engagement. Self Motivation was measured by two confidence-related items reflecting learners' belief in their programming abilities. Self Monitoring was measured by two items capturing students' perceived ability to overcome challenges and solve programming problems. Learning Strategy was measured by five items covering exam preparedness, practice frequency, real-world application, and overall course success. Academic Performance was operationalised using students' official end-of-semester numerical grade for the programming course, obtained from university records for both cohorts. This provided an objective outcome measure independent of student self-report, addressing a common limitation of survey-based studies.

Qualitative data from the open-ended questions were analysed using thematic analysis, following the inductive approach outlined by Braun and Clarke (2006), who define thematic analysis as a method for identifying, analysing, and reporting patterns within data. Codes were generated directly from participants' responses and progressively organised into broader themes capturing learners' experiences with SoloLearn. The resulting themes centred on motivation, engagement, and perceived challenges were subsequently interpreted alongside the quantitative PLS-SEM findings in a process of methodological triangulation. Specifically, qualitative themes were used to explain and contextualize statistical relationships identified in the structural model, such as how reported motivational factors and engagement challenges aligned with patterns observed in the quantitative data. The purpose of this triangulation was to use both qualitative and quantitative data to more accurately define relationships among the variables of interest (Creswell & Creswell, 2017), thereby grounding statistical relationships in participants' lived experiences and strengthening the overall validity of the findings.

3.6 Limitations and Future Research Directions

Though the research methodology provided useful findings, it contains some limitations. The study relied primarily on self-reported information which captures perceptions and confidence levels rather than direct measures of programming competence. While Academic Performance was measured using official university grades, other constructs such as self-motivation, self-monitoring, and learning strategy were based on student perceptions. Future research should complement survey data with additional performance-based evidence such as coding assignments and lab tasks to provide a more comprehensive assessment of programming competence development. The open-ended items in the study generated useful contextual information but the qualitative analysis was restricted in scope. Future research should implement mixed-methods studies that include interviews and focus groups to study learners' real-world experiences and their metacognitive learning approaches in detail. The combination of survey data with these approaches would create a complete picture of how SoloLearn and similar mobile learning platforms help students develop programming abilities.

A potential limitation of this study is common method bias, which may arise when predictor and outcome variables are collected using the same self-report instrument (Podsakoff et al., 2003). Several procedural measures were taken to mitigate this risk, including ensuring participant anonymity, making participation voluntary, and assuring confidentiality of responses. Importantly, the outcome construct, Academic Performance, was measured using official university grades rather than self-report, which partially addresses common method bias concerns by introducing an objective measure independent of the survey instrument.

4 Results

4.1 Response Rate

The study achieved a response rate of 77.8%, with 708 valid responses obtained from a target population of 910 students. The distribution of responses across the data collection period was as follows: Year 1 contributed 392 completed questionnaires, representing 43.1% of the total sampling frame, while Year 2 yielded 316 responses (34.7%). This high response rate surpasses commonly accepted benchmarks for survey-based research, thereby supporting the external validity and generalizability of the findings (Baruch & Holtom, 2008).

4.2 Demographic Characteristics

Of the 708 respondents, 626 (88.4%) identified as male and 82 (11.6%) as female, reflecting the existing gender composition of the department from which the sample was drawn. The age distribution indicated that the sample was predominantly composed of young adults, consistent with demographic patterns typically observed in tertiary-level computing education. The largest age group was 21–25 years (n = 419; 59.2%), followed by those aged 26–30 (n = 131; 18.5%) and 16–20 years (n = 95; 13.4%). A smaller proportion of participants were aged 31–35 (n = 49; 6.9%), and only 14 respondents (2.0%) were over the age of 35. These findings suggest a youthful population, aligning with broader enrolment trends in higher education, particularly within STEM-related disciplines.

(1) How does the use of SoloLearn contribute to the development of self-directed learning (SDL) skills in programming?

4.3 Self-Management and Engagement

Students exhibited notable SDL behaviours in app usage. A majority (50.4%) reported using SoloLearn on a weekly basis, while 24.8% used it daily. Regarding study time, 60.9% of respondents spent 2–5 hours per week on the app, and an additional 10% exceeded 6 hours weekly. These patterns suggest moderate to high engagement and intentional time management for programming practice. Students generally responded positively to this flexibility, with one noting: “It’s interactive, easy to follow, and great for learning at your own pace."

4.4 Strategic Learning and Feature Utilization

Quantitative analysis of feature engagement revealed three principal ways in which SoloLearn facilitated the development of SDL skills in programming education as depicted in Table 1. The platform’s feature architecture appears to support SDL through structured knowledge acquisition, peer-enabled metacognition, and adaptive challenge-seeking behaviors.

Table 1 Frequency Distribution of Feature Utilization

Feature	Count	%	SDL Dimension Supported
Answering Questions	401	24.7	Peer validation of knowledge
Learning Lessons	343	21.1	Structured content mastery
Challenging Others	302	18.6	Competency calibration
Writing Codes	278	17.1	Procedural skill development
Asking Questions	131	8.1	Strategic help-seeking

This distribution suggests that students actively used features aligned with goal-oriented and reflective learning behaviours, such as answering questions, practicing code, and engaging in peer challenges.

4.5 Confidence and Autonomy

The results indicate that 46.8% of respondents agreed or strongly agreed that SoloLearn increased their confidence in programming, while 46.8% reported improved confidence in solving programming problems independently. These findings are noteworthy, as self-efficacy is a key predictor of student persistence, engagement, and success in computing education. The interactive design of SoloLearn including gamified tasks, real-time feedback, and peer interaction likely played a role in enhancing learners’ confidence, particularly in a self-directed learning context where students must navigate content autonomously.

However, 27.1% and 24.9% of respondents selected neutral responses to the respective items, indicating uncertainty or minimal perceived impact. This neutrality may reflect the app’s role as a supplementary rather than transformative tool for some students, limited usage, or a misalignment with individual learning preferences. Additionally, 5.1% and 5.7% expressed disagreement, suggesting that a small group found SoloLearn ineffective in building confidence. These negative responses may be attributed to factors such as inadequate prior exposure to mobile learning, a mismatch between app content and learner proficiency, or unmet expectations for real-world programming application. In some cases, the lack of instructor support in self-paced environments could also contribute to lower perceived gains.

4.6 Perceived Learning Gains and Academic Impact

The assessment of SoloLearn’s impact on student academic development and programming skills was conducted through five questionnaire items. The items measured students’ readiness for exams and grades and their practice frequency and real-world application and overall course success. The educational value of SoloLearn received positive feedback from most students. The survey results showed that 48.7% of participants agreed or strongly agreed that the app helped them prepare better for programming exams and assignments thus proving its value as a supplementary study resource. Also, approximately 60% of students reported some degree of improvement in programming grades through SoloLearn but 30.5% gave a neutral response which might indicate inconsistent grade performance or other external factors.

A total of 41.2% of students practiced coding more often through SoloLearn than through conventional study approaches. The results show improved student engagement which serves as an essential sign of self-regulated learning behaviour. Furthermore, 38.4% of participants believed SoloLearn enhanced their skills to use programming concepts in real-world situations which is vital for students who will work on practical projects and seek employment. Additionally, 43.2% of students believed that the app played a role in their success in the programming course. The survey results suggest that SoloLearn is perceived as a useful learning tool that may support students' academic outcomes.

The study findings show that SoloLearn supports both cognitive and behavioural elements of SDL. The mobile learning platform was associated with improved practice habits and better perceived assessment performance and programming skill application which together demonstrate increased learner autonomy and perceived academic achievement.

Overall, the results demonstrate that SoloLearn is perceived to foster both cognitive and behavioural aspects of SDL. Increased practice frequency, confidence in assessments, and application of programming skills all reflect a heightened level of learner autonomy and academic performance associated with use of the mobile learning platform.

The descriptive findings show how students used SoloLearn and their views about its effect on SDL abilities yet it remains important to study the relationships between SDL dimensions and their effects on academic performance. The study used Partial Least Squares Structural Equation Modeling (PLS-SEM) to analyze the complex relationships between SDL dimensions and their impact on academic performance. This approach allowed for the simultaneous assessment of measurement validity and the testing of hypothesized pathways linking app usage patterns, motivational and metacognitive factors, learning strategies, and academic performance.

From Table 2, all AVE values exceed the 0.50 threshold and all rho_c values meet the 0.70 threshold, confirming convergent validity (Hair et al., 2019). The low Cronbach's alpha for Self Management is attributable to the known sensitivity of alpha to two-item constructs with unequal loadings; rho_c (0.781) and AVE (0.643) confirm acceptable reliability. Academic Performance is a single-item construct; rho_c and AVE are 1.000 by definition. Indicator loadings are shown in Figure 1.

Table 2 Path Coefficients for Structural Model

Construct	α	rho_a	rho_c	AVE	R^2
Learning Strategy	0.836	0.846	0.884	0.605	0.527
Self Management	0.460	0.507	0.781	0.643	-
Self Monitoring	0.798	0.807	0.908	0.831	0.638
Self Motivation	0.828	0.829	0.921	0.853	0.069
Academic Performance	-	-	1.000	1.000	0.100

Note: α = Cronbach's alpha; rho_a = composite reliability (Dijkstra-Henseler); rho_c = composite reliability; AVE = average variance extracted.

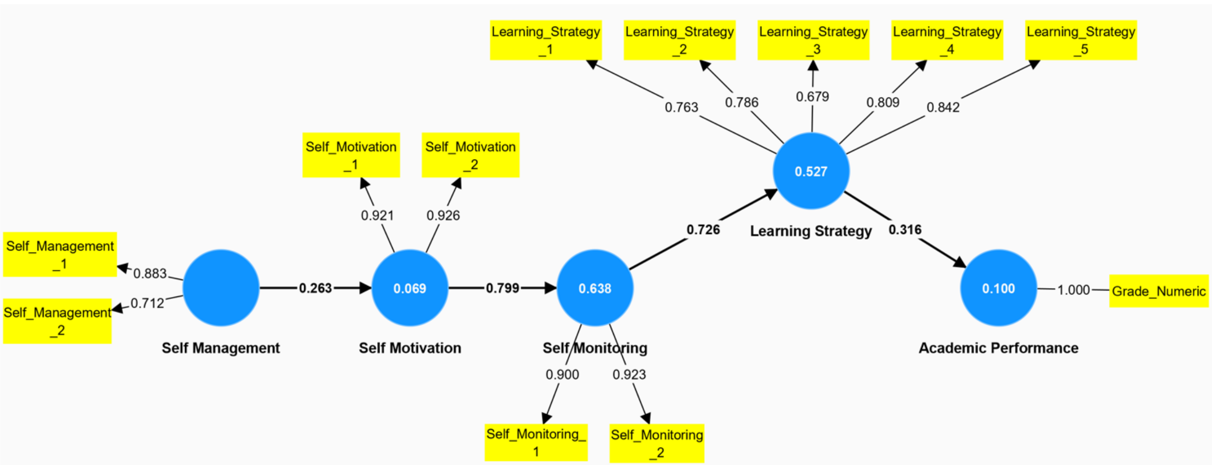

Figure 1 Structural Model

The path coefficients revealed that Self Management significantly predicted Self Motivation (β = 0.263, p < .001), indicating that students with stronger self-management skills in organising and directing their learning also reported higher motivation. Self Motivation in turn significantly predicted Self Monitoring (β = 0.799, p < .001), and Self Monitoring subsequently predicted Learning Strategy (β = 0.726, p < .001). Learning Strategy was a significant predictor of Academic Performance (β = 0.316, p < .001), suggesting that students who adopted more effective SDL strategies through SoloLearn demonstrated higher academic achievement as measured by official end-of-semester grades. It is noted that the loading for Learning_Strategy_3 (0.679) as shown in Figure 1, falls marginally below the conventional 0.708 threshold. However, as this value exceeds the acceptable minimum of 0.60 in exploratory research contexts (Hair et al., 2019), the item was retained in the model. (see Table 3)

Table 3 Path Coefficients for Structural Model

Hypothesized Path	β (Path Coefficient)	*t*-Statistic	*p*-Value
Learner Control → Self Monitoring	0.078	1.490	0.136 (ns)
Self Management → Self Motivation	0.263	6.619	0.000***
Self Motivation → Self Monitoring	0.799	39.852	0.000***
Self Monitoring → Learning Strategy	0.726	32.422	0.000***
Learning Strategy → Academic Performance	0.316	8.953	0.000***

Note: β = standardized path coefficient; ns = not significant. *** $p < 0.001$

4.7 Structural Model Findings

The structural model was evaluated using PLS-SEM to assess the relationships between key SDL dimensions and academic performance. As presented in Table 3, the model revealed several statistically significant paths. Specifically, Self Motivation had a strong and significant effect on Self Monitoring (β = 0.799, $p < 0.001$), which in turn significantly predicted Learning Strategy (β = 0.726, $p < 0.001$). Additionally, Learning Strategy positively influenced Academic Performance (β = 0.316, $p < 0.001$), suggesting that students who adopted more effective SDL strategies via SoloLearn demonstrated higher academic achievement as measured by official end-of-semester grades. Furthermore, Self Management positively impacted Self Motivation (β = 0.263, $p < 0.001$), highlighting the role of self-management skills in fostering learner motivation.

Learner Control was initially included in the structural model. However, its path to Self Monitoring was non-significant (β = 0.078, p = 0.136), and several indicator loadings fell below acceptable thresholds. Accordingly, it was removed from the final model. Feature usage patterns associated with Learner Control are reported descriptively in the RQ1 findings. These results underscore a meaningful pathway from app usage patterns to motivational and metacognitive dimensions of SDL, ultimately contributing to improved academic performance.

It is acknowledged that the R^2 values for Self Motivation (0.069) and Academic Performance (0.100) are relatively modest, indicating that these constructs are influenced by factors beyond those captured in the current model. This is consistent with the exploratory nature of the study and suggests that additional variables such as prior programming experience, instructor support, and device access may account for unexplained variance.

4.8 Discriminant Validity

Most HTMT values were below the recommended threshold of 0.90. However, the relationship between Self Motivation and Self Monitoring exceeded this threshold (HTMT = 0.977), indicating a notable measurement limitation. A value approaching 1.0 suggests that these two constructs may not be empirically distinguishable within this dataset, raising the possibility that they are capturing a single underlying dimension rather than two conceptually distinct ones. While some theoretical overlap is expected, motivation and metacognitive monitoring are recognised as interdependent processes within SDL frameworks (Henseler et al., 2014). However, the magnitude of this overlap warrants explicit acknowledgement as a study limitation. Accordingly, the structural path between Self Motivation and Self Monitoring should be interpreted with considerable caution, and findings related to this path should not be generalised without further validation using refined measurement items. Future studies should revisit the operationalisation of these constructs to establish clearer empirical boundaries between motivational and metacognitive monitoring dimensions of SDL. Other inter-construct HTMT values, such as Learning Strategy and Academic Performance (HTMT = 0.350), confirm acceptable discriminant validity for the remaining constructs in the model. (see Table 4)

Table 4 Discriminant Validity (HTMT Criterion)

	Academic Performance	Learner Control	Learning Strategy	Self Management	Self Monitoring	Self Motivation
Academic Performance						
Learner Control	0.283					
Learning Strategy	0.350	0.292				
Self Management	0.171	0.324	0.543			
Self Monitoring	0.215	0.219	0.872	0.441		
Self Motivation	0.248	0.239	0.854	0.414	0.977	

(2) What are students' perceptions of SoloLearn's usability, interactivity, and effectiveness for learning programming?

4.9 Usability and Perceived Effectiveness of SoloLearn

Students' perceptions of SoloLearn's usability were generally favourable, though not uniformly strong across all dimensions. As shown in Table 5, the highest agreement was observed for ease of use (66.2%) and user interface friendliness (67.7%), with corresponding means of 3.77 and 3.74. These results suggest that the application is largely intuitive and accessible for novice users. Similarly, ease of navigation (M = 3.50) and intuitiveness of app features (M = 3.49) were positively rated, although with more moderate agreement levels (48.6% and 49.3%, respectively). Notably, the item "speed of learning app features" received the lowest usability rating (M = 3.34), with only 44.8% agreement and a comparatively higher 16% disagreement rate, suggesting that while the interface is friendly, mastering some app functions may require additional effort.

Table 5 Students' perceptions of SoloLearn's usability, interactivity, and effectiveness for learning programming

Item	Mean	SD	% Agree/ Strongly Agree	% Disagree/ Strongly Disagree
Usability				
PEU1 (Ease of Navigation)	3.50	0.90	48.6	9.7
PEU2 (UI User Friendliness)	3.74	0.84	67.7	4.9
PEU3 (Intuitive App Features)	3.49	0.83	49.3	8.5
PEU4 (Speed of Learning App Features)	3.34	0.88	44.8	16
PEU5 (App Ease of Use)	3.77	0.71	66.2	2.7
Effectiveness				
Using the mobile app enhanced my effectiveness in learning programming.	3.81	0.71	73.7	3.8
Using the mobile app made it easier for me to understand programming concepts	3.77	0.76	68.6	4.0
I feel more capable of solving programming problems after using SoloLearn app?	3.41	0.77	46.8	8.9
SoloLearn app helped me overcome challenges in learning programming?	3.52	0.76	54	7.2
How well did the SoloLearn app prepare you for programming exams or quizzes?	3.52	0.73	48.7	3.8
How much has the app improved your ability to apply programming concepts in real-world scenarios?	3.22	0.86	38.4	19.4
To what extent has the app contributed to your overall success in programming courses?	3.33	0.88	43.2	16.5

Perceptions of SoloLearn's effectiveness for learning programming were also positive but somewhat variable across indicators. A majority of students agreed that the app enhanced their learning effectiveness (73.7%) and improved their conceptual understanding (68.6%), with mean scores of 3.81 and 3.77, respectively. These items had the highest endorsement, indicating strong perceived instructional value. Conversely, fewer students reported confidence in solving programming problems (46.8%) or overcoming learning challenges (54.0%) after using the app. Even lower levels of agreement were recorded for real-world application (38.4%) and overall course success (43.2%), with higher rates of disagreement (19.4% and 16.5%, respectively), suggesting that some students may perceive the app as more effective for theoretical reinforcement than for developing practical, transferable skills.

Taken together, these findings suggest that while SoloLearn is generally perceived as user-friendly and supportive of learning effectiveness, its impact may be stronger in facilitating conceptual understanding and basic programming practice than in advanced problem-solving, metacognitive growth, or real-world application.

4.10 Interactivity and Engagement with Application Features

Analysis of feature utilization revealed substantial engagement with SoloLearn's interactive functionalities. As presented in Table 6, 56.5% of respondents frequently engaged in answering questions, while 41.5% participated in code writing, and 40.5% were involved in peer-to-peer challenges. These activities reflect a high level of learner interaction and autonomy. In contrast, only 17.9% of students reported using the question-asking feature, and 10.0% engaged in commenting on peer content. This disparity suggests that while students actively used features supporting self-verification and practice, fewer leveraged those involving reflective or collaborative interaction. These findings highlight differences in learner behavior and comfort with public engagement tools but also underscore SoloLearn's capacity to facilitate socially embedded, inquiry-driven, and participatory learning processes within a self-directed learning framework.

Table 6 SoloLearn features used by students

Feature	% of Students Who Used Feature
Answering Questions	56.5
Writing Codes	41.5
Challenging Others	40.5
Asking Questions	17.9
Commenting	10

(3) To what extent does SoloLearn use influence students' confidence and ability to solve programming problems independently?

Students' perceptions of SoloLearn's contribution to their programming confidence and autonomy were generally positive, although with some variability. As presented in Table 7, 46.8% of respondents agreed or strongly agreed that they felt confident in their programming skills after using the app (M = 3.41, SD = 0.77). The same proportion indicated that they felt more capable of solving programming problems independently. Additionally, a slightly higher percentage (54.0%) agreed that SoloLearn helped them overcome challenges in learning programming (M = 3.52, SD = 0.76). These findings suggest that while not universally experienced, the app plays a meaningful role in supporting students' self-efficacy and perceived competence in programming tasks.

Table 7 SoloLearn's influence on students' confidence and ability to solve programming problems independently

Item	Mean	SD	% Agree/ Strongly Agree	% Disagree/ Strongly Disagree
I am confident in my programming skills after using SoloLearn.	3.41	0.77	46.8%	8.9%
I feel more capable of solving programming problems independently.	3.41	0.77	46.8%	8.9%
SoloLearn helped me overcome challenges in learning programming.	3.52	0.76	54.0%	7.2%

The reported confidence gains match the engagement patterns from the first research question because students used practice-based features like answering questions and writing code and participating in peer challenges. The repeated practice of these skills probably strengthens mastery of skills which leads to better actual and perceived competence. Together, these results indicate that SoloLearn contributes to the development of programming confidence and independence primarily by fostering self-regulated learning behaviours. While some students may not experience a transformative shift, the app appears to effectively support a substantial segment of learners in building the confidence necessary for independent problem-solving.

(4) What challenges do students face when using SoloLearn for programming education, and how can the app be improved to better support SDL?

To explore the challenges students encountered while using the SoloLearn app and how the platform can be improved to better support SDL, a thematic content analysis was conducted on open-ended responses. A total of 10 challenge categories and 9 improvement themes emerged from the data, each shedding light on specific barriers to learning autonomy, control, and motivation.

4.11 Challenges Faced in Using SoloLearn

Analysis of student feedback revealed several recurring obstacles that potentially hinder the development of SDL competencies. Table 8 presents the frequency distribution of the most cited challenge categories.

Table 8 Reported Challenges Faced When Using SoloLearn

Challenge Theme	Frequency (%)	SDL Dimension Affected
Time/Life System Limitations	32.7%	Learning continuity, motivation
Ads and Premium Restrictions	14.6%	Access to learning, autonomy
Unclear or Inconsistent Content	13.1%	Comprehension, metacognition
No Feedback/Answers for Practice Questions	9.2%	Self-monitoring, self-evaluation
Insufficient Advanced Content	8.1%	Goal setting, mastery progression
App Bugs or Technical Glitches	6.8%	Learning flow, cognitive load
Lack of Instructor/Peer Support	5.8%	Social learning, help-seeking
No Structured Curriculum Path	4.4%	Strategic learning, planning
Motivation Issues/Low Engagement	2.9%	Self-motivation
Language Barrier or Localization Issues	2.4%	Accessibility, inclusion

The Time/Life system, which restricts learning after incorrect answers unless lives are regained through waiting or payment, was the most frequently mentioned challenge. This system was viewed as disruptive to learning flow and frustrating for motivated learners, thus impeding self-pacing and sustained engagement which are both essential for SDL. Similarly, ads and premium restrictions were cited as limiting access to key content, contradicting the autonomy learners expect in self-directed platforms.

In terms of content, unclear or inconsistent explanations, particularly in coding tasks and quizzes, undermined students' confidence and ability to self-assess therefore hindering self-monitoring and strategic help-seeking. Others noted the lack of feedback on practice exercises and the absence of challenging material, which limited mastery progression and the app's utility beyond beginner-level learning.

4.12 Suggestions for Improving SoloLearn to Support SDL

Students also offered constructive suggestions for improving the platform, many of which mirrored the challenges they reported. Table 9 presents the most frequently recommended improvements.

Table 9 Suggested Improvements to SoloLearn

Suggested Improvement	Frequency (%)	SDL Principle Enhanced
Remove or Revise Time/Life System	30.2%	Learning continuity, autonomy
Reduce Ads or Make More Features Free	16.5%	Access equity, self-pacing
Improve Content Clarity and Accuracy	14.7%	Metacognition, confidence building
Provide Explanations for Practice Tasks	12.3%	Feedback, self-assessment
Add Advanced or Structured Content	9.8%	Mastery, self-directed planning
Fix Bugs and Improve App Stability	7.1%	Cognitive flow, user experience
Add Instructor or Peer Mentorship	4.6%	Help-seeking, social SDL
Include Clear Learning Paths or Roadmaps	3.6%	Goal setting, progression tracking
Improve Language Support	1.2%	Accessibility, inclusiveness

Respondents most frequently recommended removing or revising the Time/Life system, often describing it as punitive or demotivating therefore counterproductive for SDL environments where learning should be encouraged through self-correction. Others requested increased access to premium features or reduced advertisements to avoid disruptions and maintain a flow state, essential for sustained engagement in mobile learning.

The main theme focused on the need for practice question explanations and clearer content organization. The recommendations demonstrate learners' need for scaffolding tools which help them evaluate themselves and solve problems independently. Students demonstrate their need for advanced content and structured learning opportunities through their request for more challenging material.

5 Discussion

The findings of this study demonstrate that SoloLearn supports programming students in developing SDL skills through structured practice, motivational scaffolding, and strategy-enhancing features. Engagement levels were high, with 50.4% of students using the app weekly, 24.8% daily, and over 60% dedicating two to five hours per week. This suggests that mobile microlearning platforms can help address a persistent challenge in programming education which is sustaining consistent out-of-class practice. These findings align with established

SDL frameworks, which underscore the importance of autonomy, time management, and sustained engagement (Garrison, 1997; Knowles, 1975; Zimmerman, 2002). They also align with recent mobile learning research emphasizing the role of such environments in fostering time management and learner autonomy (Deng & Gao, 2023; Ifenthaler & Schumacher, 2016).

Feature utilization data revealed that students engaged more with independent practice tools such as answering questions (24.7%) and writing code (17.1%), while collaborative activities such as asking questions (8.1%) were underused. Although SoloLearn effectively supports individual mastery, its collaborative potential is not fully realized, likely due to social barriers common in competitive coding environments (Kovanović et al., 2015). Several students echoed this in their feedback, noting that while the app was "fun and easy to follow," peer support was often "slow or unhelpful." For educators, this indicates that social interaction within the platform requires intentional scaffolding rather than being left to chance.

Notably, SoloLearn's gamified features fostered self-motivation and self-monitoring, both key dimensions of SDL. Most students reported increased confidence in programming and problem-solving, attributable to badges, leaderboards, and real-time feedback. As one student put it, "earning badges made me want to keep practicing every day." These findings are consistent with research showing that gamification enhances intrinsic motivation by supporting competence and autonomy (Ryan & Deci, 2017; Sailer & Homner, 2019). Still, 20–30% of learners gave neutral responses, reflecting differences in background, workload, and learning preferences (Ifenthaler & Yau, 2020). This highlights the need for personalization to accommodate diverse motivational profiles. These neutral or negative perceptions can be interpreted through cognitive load theory (Sweller, 1988), whereby fragmented content, advertisements, and the time/life system may interrupt learning flow and increase extraneous cognitive load, thereby reducing perceived effectiveness. Additionally, the limited availability of structured learning pathways and advanced content suggests insufficient scaffolding essential for supporting learner progression from novice to advanced programming competence (Wood et al., 1976).

The structural model further clarified the SDL process underlying these outcomes. The sequential pathway Self-Management → Self-Motivation → Self-Monitoring → Learning Strategy → Academic Performance provides a roadmap for instructional integration. This sequential pathway mirrors the model of self-regulated learning of Zimmerman (2002), where motivation drives monitoring and strategy development, and resonates with Garrison (1997) emphasis on self-management and reflection as prerequisites for deep learning outcomes. The positive link between self-management and motivation ($\beta = 0.263$) suggests that consistent app use builds confidence; instructors could encourage this through low-stakes, regular SoloLearn tasks (e.g., one module per week). The strong path from motivation to monitoring ($\beta = 0.791$) highlights the role of confidence in effective progress tracking; educators can amplify this by connecting SoloLearn's progress badges to course milestones (e.g., using the Java Arrays badge as a readiness indicator for quizzes). The positive relationship between learning strategies and academic performance ($\beta = 0.316$) indicates that app-based strategies contribute to tangible outcomes, which could be reinforced by classroom debriefs and reflective exercises. By contrast, the non-significant learner control path ($\beta = 0.078$) shows that feature abundance does not automatically lead to effective monitoring; students benefit from explicit guidance on how to use features strategically, such as tutorials on the "Challenge" mode for self-testing.

This study contributes to global debates on mobile self-directed learning in computing education by demonstrating that while mobile coding platforms support foundational SDL processes such as motivation, monitoring, and structured practice, they remain limited in supporting higher-order skills such as adaptive problem-solving and real-world application. This highlights a critical gap between engagement-focused mobile learning design and the development of advanced programming competence. From a developmental perspective, these findings suggest that such platforms are most effective at supporting early-stage programming skills, including syntax familiarity and routine problem-solving. However, progression toward advanced competencies such as debugging complex systems, developing full-scale applications, and applying concepts in authentic contexts requires additional scaffolding, structured learning pathways, and opportunities for project-based and collaborative learning. This indicates that while mobile apps can initiate self-directed learning, sustained skill development depends on their integration with broader instructional and experiential learning environments.

5.1 Perceptions of Usability, Effectiveness, and Interactivity

Students rated SoloLearn highly for usability, citing its clear navigation and simple interface, consistent with mobile learning usability heuristics (Kukulska-Hulme et al., 2023). "It's very easy to get started," one student noted, although others complained that "too many ads interrupt learning." Usability challenges emerged when using advanced features, echoing the observation of Ifenthaler and Yau (2020) that learners require scaffolding when transitioning from novice to expert tool use. Adaptive tutorials and context-sensitive tooltips could address onboarding issues (Roll et al., 2021).

SoloLearn proved effective for building conceptual knowledge and test readiness, supporting findings that gamified microlearning enhances retention in STEM fields (Giannakos et al., 2022; Sailer & Homner, 2019). However, students expressed difficulty transferring skills to real-world contexts: "It helps for quizzes but not for big projects." This indicates a need for project-based tasks and authentic problem scenarios to extend practice beyond drills and into meaningful application (Zheng et al., 2026).

5.2 Confidence, Independence, and Problem-Solving

SoloLearn fostered confidence and independent problem-solving, consistent with Bandura (1997) self-efficacy theory. Gamified elements such as badges and progress tracking boosted motivation by fulfilling needs for competence and autonomy (Ryan & Deci, 2017). Yet 20-30% of learners remained neutral, reflecting differences in background and workload management (Ifenthaler & Yau, 2020). This suggests that standardized platform designs cannot fully meet the needs of diverse learner profiles (Koedinger et al., 2023).

The findings also revealed a gap between routine and adaptive expertise. SoloLearn excels at routine expertise like supporting structured and repetitive practice; however, it offers limited support for adaptive problem-solving needed in real-world contexts (Faber et al., 2024). Suggested improvements included peer debugging tasks, self-explanation prompts, and adaptive scaffolding, aligning with recent calls for platforms to foster both individual agency and social collaboration (Aleven et al., 2016; Zheng et al., 2026).

5.3 Instructional Design Implications

The research findings demonstrate that SoloLearn functions best as part of a blended instructional approach rather than as a standalone tool. Educators can integrate the app into their practice in four key ways:

(1) Assign SoloLearn modules as preparatory homework or low-stakes graded tasks, allowing flexible integration across different course sizes and delivery modes.

(2) Integrate SoloLearn achievements (e.g., badges or completion milestones) into course assessment structures to incentivize sustained engagement.

(3) Combine app-based drills with lab assignments and mini-projects to bridge foundational practice with real-world programming tasks.

(4) Incorporate collaborative peer activities, such as group debugging exercises or discussion-based challenges, to enhance social learning and scalability across cohorts.

(5) Require short reflection reports or learning logs after module completion to support metacognitive development and self-regulated learning.

While these instructional strategies are applicable within the context of this study, their implementation may vary across institutional and cultural settings. Differences in technological infrastructure, internet accessibility, class sizes, and pedagogical traditions may influence how effectively mobile learning tools like SoloLearn can be integrated. For example, institutions with limited connectivity may find continuous app-based engagement challenging, while highly exam-oriented systems may require alignment with formal assessment structures. Additionally, cultural factors such as learner autonomy, collaboration norms, and attitudes toward self-directed learning may shape student engagement with mobile platforms. These considerations highlight the need for context-sensitive adaptation when scaling such interventions across diverse educational environments.

5.4 Challenges and Improvement Strategies

The students identified multiple limitations which restrict the long-term effectiveness of SoloLearn as a self-directed learning platform. The main drawback of SoloLearn is its limited content depth because learners found little instructional value in advanced concepts. The research confirms that microlearning models face criticism for their inability to connect the intermediate gap in programming education (Koedinger et al., 2023; Wang et al., 2023).

Help-seeking challenges were also prominent. The community support features of SoloLearn suffered from delayed and unhelpful responses which reduced their educational worth. The problems with unmoderated online peer support in coding education match the issues reported by Kovanović et al. (2018). The learning process faced obstacles due to usability problems which stemmed from mobile code editing difficulties and inconsistent feature performance and the absence of offline capabilities (Alrasheedi et al., 2016).

Curricular rigidity was another barrier. The platform's linear content progression failed to account for learners' prior knowledge or pacing preferences, contradicting adaptive learning principles (Ifenthaler & Schumacher, 2016). Current research supports personalized pathways, informed by pre-assessment and real-time performance data, as more effective in promoting mastery (Aleven et al., 2022).

5.5 Platform and Research Implications

Beyond immediate instructional practices, this study highlights opportunities for platform development and further research. Addressing the challenges identified by students such as limited content depth, weak peer support, and usability constraints requires innovations in mobile coding platforms. Future work should explore how enhancements like authentic project-based modules, AI-curated feedback, and adaptive learning trajectories can improve outcomes across diverse institutions and cultural contexts, ensuring broader applicability of mobile learning tools in programming education.

6 Conclusion

This study demonstrates how mobile microlearning platforms like SoloLearn can foster SDL in programming education by promoting time management, confidence, and strategy use. Engagement patterns and structural modeling results show that consistent app use builds motivation, which in turn supports monitoring, learning strategies, and perceived academic performance. The findings further highlight design gaps. While SoloLearn is effective for routine expertise and conceptual knowledge, it provides limited support for advanced application, adaptive expertise, and collaboration.

Beyond SoloLearn, the results inform the design of blended and mobile pedagogies in computing education. Specifically, integrating gamified modules, reflective prompts, scaffolded peer interaction, and project-based tasks can enhance the effectiveness of mobile learning in programming. These insights extend current mobile learning scholarship by situating SDL processes within the unique challenges of programming instruction.

At the same time, the scope of this study is bounded by its context. The sample, drawn from a single institution and largely male-dominated, reflects common patterns in computing education but limits the representativeness of underrepresented groups. Institutional culture and regional factors may also shape adoption and engagement in ways that differ across settings. Future research should therefore examine more gender-diverse cohorts, multiple institutions, and cross-cultural contexts to clarify the broader relevance and inclusivity of mobile SDL platforms.

6.1 Limitations

The research faces multiple constraints that need to be considered. Self-report measures create potential biases because they cannot verify actual skill acquisition through objective assessments such as coding tests (Broadbent & Poon, 2015; Podsakoff et al., 2003). The cross-sectional study design prevents researchers from studying the non-linear patterns that characterize programming mastery development (Margulieux et al., 2019); longitudinal approaches are needed to capture sustained outcomes. The scope of the study is also limited, as participants were drawn from a single region, where cultural and motivational elements may shape engagement (Nye et al., 2018). The brief open-ended responses restricted the collection of qualitative insights; richer approaches such as think-aloud protocols or experience sampling would have provided more detailed process information (D'Mello et al., 2014; Taub et al., 2022). Finally, the study did not control for potential moderators such as prior programming experience, device access, and usage intensity, which require further investigation (Crompton et al., 2021; Gašević et al., 2017; Prinsloo et al., 2024).

6.2 Generalizability of Findings

The generalizability of this study is limited by its single-institution context and male-dominated sample (88.4% male). While this gender distribution reflects common patterns in computing education in Ghana and similar contexts, it significantly restricts insights into the experiences of female learners and other underrepresented groups. Prior research highlights that gender shapes persistence, confidence, and engagement in computing differently, with female students often reporting lower self-efficacy and facing additional barriers in technology-oriented environments. The findings of this study should therefore not be assumed to apply equally across gender groups. Future studies should deliberately recruit more gender-diverse and multi-institutional samples to examine whether the SDL pathway identified here holds across different learner populations, and to surface any gender-specific patterns in mobile learning engagement and outcomes.

Ethics and Informed Consent

The study was conducted in compliance with ethical guidelines and was formally approved by the Institutional Ethics Committee of the University of Education, Winneba. All participants were informed about the purpose and procedures of the study, and participation was entirely voluntary. Informed consent was obtained before data collection, and confidentiality and anonymity were assured throughout the process. Completion and submission of the online questionnaire served as confirmation of consent to participate.

Conflicts of Interest

The author declares that he has no conflicts of interest.

References

Akter, S., Fosso Wamba, S., & Dewan, S. (2017). Why PLS-SEM is suitable for complex modelling? An empirical illustration in big data analytics quality. Production Planning &Amp; Control, 28(11-12), 1011–1021.
https://doi.org/10.1080/09537287.2016.1267411

Aleven, V., Blankestijn, J., Lawrence, L., Nagashima, T., & Taatgen, N. (2022). A Dashboard to Support Teachers During Students' Self-paced AI-Supported Problem-Solving Practice. Educating for a New Future: Making Sense of Technology-Enhanced Learning Adoption, 16–30.
https://doi.org/10.1007/978-3-031-16290-9_2

Aleven, V., McLaren, B. M., Sewall, J., van Velsen, M., Popescu, O., Demi, S., Ringenberg, M., & Koedinger, K. R. (2016). Example-tracing tutors: intelligent tutor development for non-programmers. International Journal of Artificial Intelligence in Education, 26(2), 224-269.
https://doi.org/10.1007/s40593-015-0088-2

Alrasheedi, M., Capretz, L. F., & Raza, A. (2016). A systematic review of the critical factors for success of mobile learning in higher education (university students' perspective). Computers in Human Behavior, 63, 538–544.

Amro, J. S., & Romli, R. (2019). Investigation on the Learning Programming Techniques via Mobile Learning Application. 2019 4th International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE), 1–7.
https://doi.org/10.1109/icraie47735.2019.9037764

Bandura, A. (1997). Self-efficacy: The exercise of control. W. H. Freeman.

Baruch, Y., & Holtom, B. C. (2008). Survey response rate levels and trends in organizational research. Human Relations, 61(8), 1139–1160.
https://doi.org/10.1177/0018726708094863

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. Qualitative Research in Psychology, 3(2), 77–101.
https://doi.org/10.1191/1478088706qp063oa

Broadbent, J., & Poon, W. L. (2015). Self-regulated learning strategies & academic achievement in online higher education learning environments: A systematic review. The Internet and Higher Education, 27, 1-13.
https://doi.org/10.1016/j.iheduc.2015.04.007

Calderón-Garrido, D., Ramos-Pardo, F. J., & Suárez-Guerrero, C. (2022). The Use of Mobile Phones in Classrooms: A Systematic Review. International Journal of Emerging Technologies in Learning (iJET), 17(06), 194–210.
https://doi.org/10.3991/ijet.v17i06.29181

Calderon, I., Silva, W., & Feitosa, E. (2023). Active Learning Methodologies for Teaching Programming in Undergraduate Courses: A Systematic Mapping Study. Informatics in Education.
https://doi.org/10.15388/infedu.2024.11

Creswell, J. W., & Creswell, J. D. (2017). Research design: Qualitative, quantitative, and mixed methods approaches. Sage publications.

Crompton, H., Burke, D., Jordan, K., & Wilson, S. W. G. (2021). Learning with technology during emergencies: A systematic review of K-12 education. British Journal of Educational Technology, 52(5), 1554–1574.
https://doi.org/10.1111/bjet.13114

Dahri, N. A., Al-Rahmi, W. M., Almogren, A. S., Yahaya, N., Vighio, M. S., Al-maatuok, Q., Al-Rahmi, A. M., & Al-Adwan, A. S. (2023). Acceptance of Mobile Learning Technology by Teachers: Influencing Mobile Self-Efficacy and 21st-Century Skills-Based Training. Sustainability, 15(11), 8514.
https://doi.org/10.3390/su15118514

Deng, R., & Gao, Y. (2023). Using learner reviews to inform instructional video design in MOOCs. Behavioral Sciences, 13(4), 330.
https://doi.org/10.3390/bs13040330

D'Mello, S. K., Lehman, B., Pekrun, R., & Graesser, A. (2014). Confusion can be beneficial for learning. Learning and Instruction, 29, 153–170.
https://doi.org/10.1016/j.learninstruc.2012.05.003

Faber, T. J. E., Dankbaar, M. E. W., van den Broek, W. W., Bruinink, L. J., Hogeveen, M., & van Merriënboer, J. J. G. (2024). Effects of adaptive scaffolding on performance, cognitive load and engagement in game-based learning: a randomized controlled trial. BMC Medical Education, 24(1).
https://doi.org/10.1186/s12909-024-05698-3

Fredricks, J. A., Blumenfeld, P. C., & Paris, A. H. (2004). School Engagement: Potential of the Concept, State of the Evidence. Review of Educational Research, 74(1), 59–109.
https://doi.org/10.3102/00346543074001059

Gašević, D., Siemens, G., & Rosé, C. P. (2017). Guest editorial: Special section on learning analytics. IEEE Transactions on Learning Technologies, 10(1), 3-5.

Garrison, D. R. (1997). Self-Directed Learning: Toward a Comprehensive Model. Adult Education Quarterly, 48(1), 18–33.
https://doi.org/10.1177/074171369704800103

Giannakos, M., Cukurova, M., & Papavlasopoulou, S. (2022). Sensor-based analytics in education: Lessons learned from research in multimodal learning analytics. In The multimodal learning analytics handbook (pp. 329-358). Cham: Springer International Publishing.

Hair, J. F., Hult, G. T. M., Ringle, C. M., Sarstedt, M., Danks, N. P., & Ray, S. (2021). Partial Least Squares Structural Equation Modeling (PLS-SEM) Using R. In (Editor), Classroom Companion: Business. Springer International Publishing.
https://doi.org/10.1007/978-3-030-80519-7

Hair, J. F., Risher, J. J., Sarstedt, M., & Ringle, C. M. (2019). When to use and how to report the results of PLS-SEM. European Business Review, 31(1), 2–24.
https://doi.org/10.1108/ebr-11-2018-0203

Henseler, J., Ringle, C. M., & Sarstedt, M. (2014). A new criterion for assessing discriminant validity in variance-based structural equation modeling. Journal of the Academy of Marketing Science, 43(1), 115–135.
https://doi.org/10.1007/s11747-014-0403-8

Ifenthaler, D. (2021). Learning analytics for school and system management. OECD Digital Education Outlook, 161.

Ifenthaler, D., & Schumacher, C. (2016). Student perceptions of privacy principles for learning analytics. Educational Technology Research and Development, 64(5), 923-938.
https://doi.org/10.1007/s11423-016-9477-y

Ifenthaler, D., & Yau, J. Y. K. (2020). Towards comparative learning analytics: A framework for analysing digital learning traces across contexts. Educational Technology Research and Development, 68(3), 1027-1049. https://doi.org/10.1007/s11423-020-09788-z

Knowles, M. S. (1975). Self-directed learning (Vol. 291). New York: association press.

Koedinger, K. R., Carvalho, P. F., Liu, R., & McLaughlin, E. A. (2023). An astonishing regularity in student learning rate. Proceedings of the National Academy of Sciences, 120(13).
https://doi.org/10.1073/pnas.2221311120

Kovanović, V., Joksimović, S., Gašević, D., Siemens, G., & Hatala, M. (2015). What public media reveals about MOOCs: A systematic analysis of news reports. British Journal of Educational Technology, 46(3), 510-527.
https://doi.org/10.1111/bjet.12277

Kovanović, V., Joksimović, S., Poquet, O., Čukić, I., Gašević, D., Dawson, S., & Siemens, G. (2018). Exploring development of social capital in a CMOOC through language and discourse. Internet and Higher Education, 36, 54-64.
https://doi.org/10.1016/j.iheduc.2017.09.004

Kukulska-Hulme, A., Bossu, C., Charitonos, K., Coughlan, T., & Ferguson, R. (2023). Innovating Pedagogy 2023: Open University Innovation Report 11. The Open University.

Margulieux, L. E., Morrison, B. B., & Decker, A. (2019). Design and pilot testing of subgoal labeled worked examples for five core concepts in CS1. In Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education (pp. 31-37).
https://doi.org/10.1145/3304221.3319756

Nye, B. D., Pavlik, P. I., & Olney, A. M. (2018). Help-seeking sequences in open-ended tutoring activities. In Proceedings of the 11th International Conference on Educational Data Mining (EDM) (pp. 212–219).

Oyelere, S. S., Suhonen, J., Wajiga, G. M., & Sutinen, E. (2017). Design, development, and evaluation of a mobile learning application for computing education. Education and Information Technologies, 23(1), 467–495.
https://doi.org/10.1007/s10639-017-9613-2

Prinsloo, P., Khalil, M., & Slade, S. (2024). Learning analytics as data ecology: a tentative proposal. Journal of Computing in Higher Education, 36, 154–182.
https://doi.org/10.1007/s12528-023-09355-4

Podsakoff, P. M., MacKenzie, S. B., Lee, J.-Y., & Podsakoff, N. P. (2003). Common method biases in behavioral research: A critical review of the literature and recommended remedies. Journal of Applied Psychology, 88(5), 879–903.
https://doi.org/10.1037/0021-9010.88.5.879

Roll, I., Russell, D. M., & Gašević, D. (2021). Analytics-based adaptive scaffolding for self-regulated learning at scale. In LAK21: 11th International Conference on Learning Analytics and Knowledge (pp. 198–207). ACM.

Ryan, R. M., & Deci, E. L. (2017). Self-determination theory: Basic psychological needs in motivation, development, and wellness. Guilford Press.

Sailer, M., & Homner, L. (2019). The Gamification of Learning: a Meta-analysis. Educational Psychology Review, 32(1), 77–112.
https://doi.org/10.1007/s10648-019-09498-w

Salleh, A. M., Desa, M. M., & Tuit, R. M. (2013). The Relationship between the Learning Ecology System and Students' Engagement: A Case Study in Selangor. Asian Social Science, 9(12).
https://doi.org/10.5539/ass.v9n12p110

Schnieder, M., & Williams, S. (2023). Educational Mobile Apps for Programming in Python: Review and Analysis. Education Sciences, 13(1), 66.
https://doi.org/10.3390/educsci13010066

Shin, W. S., & Kang, M. (2015). The use of a mobile learning management system at an online university and its effect on learning satisfaction and achievement. The International Review of Research in Open and Distributed Learning, 16(3).
https://doi.org/10.19173/irrodl.v16i3.1984

Spirina, Ye. A. (2024). Interactive mobile platforms for programming training: Effect on personnel training for the IT industry. Electronic Library of Belarusian National Technical University.

Sweller, J. (1988). Cognitive Load During Problem Solving: Effects on Learning. Cognitive Science, 12(2), 257–285. Portico.
https://doi.org/10.1207/s15516709cog1202_4

Taub, M., Azevedo, R., Rajendran, R., Cloude, E. B., Biswas, G., & Price, M. J. (2022). How are students' emotions related to the accuracy of cognitive and metacognitive processes during learning with an intelligent tutoring system? Learning and Instruction, 84, 101734.
https://doi.org/10.1016/j.learninstruc.2019.04.001

Wang, X., Liu, Q., Pang, H., Tan, S. C., Lei, J., Wallace, M. P., & Li, L. (2023). What matters in AI-supported learning: A study of human-AI interactions in language learning using cluster analysis and epistemic network analysis. Computers &Amp; Education, 194, 104703.
https://doi.org/10.1016/j.compedu.2022.104703

Wold, H. (1982). Soft modeling: the basic design and some extensions. Systems under indirect observation, Part II, 2, 36-37.

Wood, D., Bruner, J. S., & Ross, G. (1976). THE ROLE OF TUTORING IN PROBLEM SOLVING *. Journal of Child Psychology and Psychiatry, 17(2), 89–100. Portico.
https://doi.org/10.1111/j.1469-7610.1976.tb00381.x

Zheng, G., Wang, Y., & Du, J. (2026). A dual-path framework for enhancing student engagement and learning outcomes in sports education: Integrating technology acceptance, self-regulation, and self-efficacy. PLoS One, 21(3), e0345809.
https://doi.org/10.1371/journal.pone.0345809

Zimmerman, B. J. (2002). Becoming a Self-Regulated Learner: An Overview. Theory into Practice, 41(2), 64–70.
https://doi.org/10.1207/s15430421tip4102_2