Unlocking the Secrets of Student Success in Low-Code Platforms: An In-Depth Comparative Analysis

Prof. Mariza Tsakalerou, Nazarbayev University

Dr. Tsakalerou is an Associate Professor at the School of Engineering and Digital Sciences at Nazarbayev University (NU) and the Director of the Engineering Management Programs. NU has debuted in the top 30 percent of international research universities in the Times Higher Education (THE) World University Rankings 2024. Dr. Tsakalerou is an active researcher, practitioner, and academic in innovation management, technology networks, and engineering education. Her current research interests focus on the relationship between group learning modalities and creativity performance. Dr. Tsakalerou is active in international forums (such as the OECD Idea Factory, the European Innovation Summit, the Joint Institute for Innovation Policy, and the European Higher Education Society) and a contributor to the Asia-Europe Foundation's Education Hub.

Michalis N. Xenos, University of Patras
Ms. Semira Maria Evangelou, University of Patras

I hold a diploma (5 years Engineering degree with an Integrated M.Sc.) from the Department of Computer Engineering & Informatics at the University of Patras. I also completed a master's degree in the Human-Computer Interaction field. During my master's studies, I discovered my passion for UX design and the importance of that field. I am software engineering, who loves exploring complex projects in search of delightful user experiences. I am currently a Ph.D. candidate at the University of Patras and my research interests are mainly focused on Human-Computer Interaction (HCI).

Unlocking the Secrets of Student Success in Low-Code Platforms: An In-Depth Comparative Analysis

Mariza Tsakalerou[a], Semira Maria Evangelou[b], Michalis Xenos[b]
[a]*School of Engineering and Digital Sciences, Nazarbayev University, Kazakhstan*
[b]*Computer Engineering and Informatics Department, University of Patras, Greece*

Abstract: This paper presents a comprehensive exploration of students' experiences with low-code platforms, which emphasize high-level logic and functionality over intricate coding. This is achieved by utilizing pre-test and post-test questionnaires, Mendix Studio Pro as an exemplar of such platforms, and structured experimentation tasks. The pre-test captures participants' baseline perspectives, while the post-test examines shifts after hands-on interaction. The study addresses research questions concerning demographic correlations, coding backgrounds, attitude changes, coding performance, and qualitative insights. The experiment protocol includes an instructional video, demographic surveys, and three tasks. Descriptive statistics and statistical tests provide insights. Performance discrepancies between IT and non-IT backgrounds are statistically significant. Feedback indicates positive perceptions of low code.

1. Introduction

In recent years, the intersection of technology and education has undergone a profound transformation, with emerging paradigms reshaping traditional approaches to teaching and learning. One such paradigm that has garnered increasing attention is low-code development—a revolutionary approach to software creation that empowers individuals, regardless of their technical background, to design and deploy fully functional applications with minimal coding expertise. Low-code platforms provide intuitive interfaces, pre-built components, and drag-and-drop functionalities, significantly reducing the barriers to entry into the world of software development. The integration of low-code development into academic curricula represents a pivotal advancement in educational practices. As technology becomes increasingly ubiquitous across industries, the ability to navigate and harness its potential has become a vital skill for students across diverse disciplines.

By simplifying the development process, low-code platforms complement existing educational tools, enhancing students' abstraction skills and computational thinking. Furthermore, low-code platforms foster innovation, creativity, and entrepreneurship among students, empowering them to develop solutions to real-world problems and make meaningful contributions to society. the integration of low-code development into academic curricula represents a transformative shift in education, offering students new opportunities to engage with technology and develop practical skills in software development.

Low code finds extensive use across various industries, including academia, where it empowers students and professionals without extensive IT backgrounds to create custom applications and can equip learners with practical, hands-on experience in application development, preparing them for the demands of a rapidly evolving digital landscape. By eliminating the need for intricate coding skills, low code platforms democratize software development, making it accessible to a broader audience. Popular low code platforms such as Microsoft Power Apps, Salesforce Lightning, and Mendix offer diverse features and customization options, catering to the specific needs of academia and beyond.

The Magic Quadrant for Enterprise Low-Code Application Platforms serves as a comprehensive analysis and evaluation of the evolving landscape of low-code development. This report is a valuable resource for enterprises seeking insights into the market's leading players, their performance, and the strategic direction of the low-code application platform (LCAP) space. Among the distinguished leaders within this domain, Mendix shines prominently, acknowledged for its exemplary prowess in low-code innovation and solution delivery [1].

2. Literature Review

Low-code development platforms are cloud-based software tools that empower developers with varying levels of expertise to create fully functional applications [2]. They leverage model-driven engineering principles, cloud infrastructure, and automatic code generation to streamline application development [3]. These platforms capitalize on cloud computing technologies like Platform-as-a-Service (PaaS) to ensure efficient development,

deployment, and maintenance of applications. Embracing model-driven engineering principles, low-code platforms harness automation and abstraction capabilities.

Moreover, Low-code platforms emerge as an advancement in making coding more accessible and aligning with broader educational initiatives. These platforms, exemplified by the likes of Mendix, embody a revolutionary shift in software engineering, prioritizing high-level logic and functionality over intricate coding. This approach not only democratizes software development by enabling contributions from non-specialist users but also resonates with the pedagogical goal of fostering computational thinking among students [4]. By simplifying the development process, low-code platforms can serve as an extension of established educational tools like block-based coding environments, which are designed to enhance students' abstraction skills and computational thinking.

In recent years, an extensive body of research literature has emerged, collectively highlighting the transformative potential of low-code platforms. These platforms serve as powerful equalizers in the context of artificial intelligence and software development, offering tools for sustainable global development and democratizing software creation for individuals outside the traditional computer science sector. Current research encapsulates a breadth of studies that underscore the revolutionary nature of low-code platforms in democratizing technology creation and advancing global development efforts [3]. Emerging as conduits for inclusive innovation, these platforms offer a pragmatic approach to engage a more diverse demographic in software development and AI [4].

The research presented by [5] provides a cornerstone for understanding the potential of low-code platforms to expand the reach of AI to non-experts. This enables a participatory approach to solving critical global issues, leveraging the collective intelligence of a wider audience. It suggests that these platforms can serve as catalysts for empowerment, enabling individuals across various domains to contribute to the conversation and solution of pressing global challenges. Parallel to these studies, research [6] investigates the educational sphere, illustrating how low-code platforms like OutSystems and Mendix are instrumental in bridging the theoretical and practical divide in software engineering education [6]. This approach advocates for an experiential learning environment where students are not mere consumers of knowledge but active participants in a simulated professional setting. It highlights the alignment of academic curricula with industry demands, equipping students with a more robust and relevant skill set for the digital economy.

It has been also investigated [7] whether the frontier of programming language development, that aims to create tools is intuitive and accessible for citizen developers. This approach enhances the inclusivity narrative by positing a future where the barrier to entry into software development is significantly lowered, allowing individuals with little to no formal training in coding to actively contribute to the creation of software solutions. Moreover, the exploration into low-code development within the construction industry, as presented by studies conducted in 2023 [8] offers a granular perspective on the implementation of these platforms in a traditionally non-digital sector. This work is seminal in discussing the operational efficiencies and innovative prospects afforded by low-code platforms, as well as addressing the potential drawbacks that may arise from an over-dependence on said platforms.

At the same time, another work [9] that takes a multidisciplinary approach provides a retrospective view of the evolution of low-code platforms, elucidating their strategic integration with ERP systems. It reflects on the historical progression from model-driven development to the current state where low-code platforms are essential in enhancing business processes, fostering agility, and enabling customization in enterprise solutions. The collective insights create a strong narrative that not only enriches the understanding of low-code/no-code platforms' educational value but also their practical implications across industries. Low-code platforms contribute to a broader discourse on how technologies are reshaping the future of work, education, and global development while elucidating student success in low-code platforms.

In a 2023 research project conducted in Sweden [10] significant findings were revealed on the impact of targeted educational strategies on code quality in early computer science education. These underscore the value of studies examining the effects of various demographic and educational factors on low-code development performance and indicate that specific pedagogical focus can lead to substantial improvements in technical competencies. Such studies could illuminate pathways to enhance educational approaches in low-code curricula, optimizing the skillsets required for contemporary software development landscapes. The implementation [11] of a low-code method in the creation of a higher education statistical reports system highlights the effectiveness of this approach in streamlining complex data management tasks. This also accentuates the need for a study to investigate how different demographic and educational backgrounds influence the utilization and performance of individuals in low-code development environments, potentially unlocking further efficiencies in data-driven administrative processes.

Past studies [12] highlight the significant role of cross-cultural experiences in enhancing engineers' global competencies, emphasizing the need for courses that integrate intercultural and global awareness into engineering curricula. This also aligns with the growing emphasis on low-code platforms in engineering education, suggesting that incorporating global competency development into low-code learning environments could prepare students

for the diverse challenges of the global engineering landscape, fostering more inclusive and culturally aware engineering solutions.

The literature review highlights the impact of low-code platforms not only in engineering education but also in their use across various industries, pointing towards a future of inclusive technology creation. It underscores the educational benefits of these platforms, fostering a hands-on learning experience that aligns with industry needs and prepares students for the digital economy. Henceforth, this work investigates the different factors that can affect the student's success in low-code platforms. From the theoretical backdrop to the practical application, the methodology will work into the empirical examination of these platforms. Utilizing pre- and post-questionnaires, it aims to capture and analyze the participants' experiences with the Mendix low-code platform, estimating shifts in perspectives and competencies that these educational tools engender.

This article has been organized as follows. Section 2 initiates with a review of relevant literature, providing essential background knowledge. Following this, Section 3 introduces the methodology employed, outlining the steps necessary for conducting the research. In Section 4, the research questions are presented and discussed in detail. Section 5 extends the discussion further, exploring additional perspectives or implications arising from the research. Finally, Section 6 concludes the study, summarizing the key findings and potential avenues for future research.

3. Materials and Tools

3.1 Pre-Post Questionnaire

Pre-test and post-test questionnaires were used to gain insights into the participants' perspectives and experiences with the Mendix low-code platform. The pre-test questionnaire served as a baseline assessment, capturing participants' demographic details, prior programming experiences, knowledge levels, and initial attitudes toward low-code development. This set of questions aimed to establish a foundational understanding of the participants before engaging with the low-code platform. On the other hand, the post-test questionnaire allowed for a reflective examination of how participants' attitudes, experiences, and perceptions shifted after hands-on interaction with the platform. By revisiting key aspects, such as attitudes toward low-code development, perceived difficulty, and the impact on application development speed, the post-test questionnaire facilitates the analysis of the participants' journey.

3.2 Mendix Studio Pro

In this experiment, Mendix Studio Pro version 10.5.1 was used as the low code tool. Mendix is a low-code development platform that empowers users to create web and mobile applications with minimal coding effort. It provides a visual development environment, allowing users to design, build, and deploy applications efficiently. Mendix Studio Pro is a specific edition of the Mendix platform tailored for professional developers, offering advanced capabilities for more intricate application development. With Mendix Studio Pro, users can leverage a range of features to design complex data models, create responsive user interfaces, and implement custom logic through microflows. Version 10.5.1 represents a specific release of the software, indicating the iterative improvements and updates made to enhance functionality and user experience. Using Mendix Studio Pro in this experiment allowed participants to engage with a sophisticated low-code development environment, contributing to a deeper understanding of their experiences and competencies within this technological framework.

4. Research Questions

This work explores students' experiences and perceptions of a low-code platform, contributing valuable insights to the ongoing discourse on low-code development within diverse educational settings. In the context of this work, the following set of Research Questions (RQs) was addressed:

- RQ1: Do statistically significant differences exist in participants' initial attitudes towards low-code platforms when considering various demographic factors, including gender identity, academic qualifications, and current country of residence?
- RQ2: Are there statistically significant differences in participants' experiences with the low-code platform when considering various demographic factors, including gender identity, academic qualifications, and current country of residence?
- RQ3: How does participants' perception of the difficulty of low-code development change after interacting with the low-code platform?

- RQ4: Are there significant differences in performance scores between participants with distinct coding backgrounds, specifically comparing those with IT backgrounds to those without, in the low-code development tasks?

5. Experiment Protocol

5.1 Participants

Overall, 28 participants took part in this study with diverse academic backgrounds were gathered. Enrolled in master's degree programs, they spanned fields like Mathematics, Economics, Engineering, and Business Administration, reflecting varied expertise levels in programming and software development. Notably, a substantial portion identified as beginners (18), some as intermediate (3), and a few as advanced (2), with 5 participants having no prior experience. Queries about familiarity with databases showcased a spectrum, emphasizing the inclusive survey nature.

Regarding low code, participants exhibited limited pre-existing knowledge, highlighting the video's potential role in enhancing understanding. Despite the diverse perspectives on the challenge posed by low-code for non-IT backgrounds, participants generally held a positive attitude. A consensus emerged on the positive impact of low-code on development speed, aligning with recognized benefits in streamlining the application development lifecycle In this study, the interaction with the Mendix low-code platform is analyzed through a structured experimentation process. This research aims to comprehensively explore the dynamics shaping students' experiences with the Mendix low-code platform. The overarching goal of this research is to identify factors that are correlated with low-code performance, experience, and attitude.

5.2 Introductory Video to Mendix

The study begins with the introduction of participants to the Mendix platform through an instructional video. This video aims to provide a basic understanding of Mendix and establish foundational knowledge among participants.

5.3 First Questionnaire

Following the instructional video, a prequestionnaire collects data on participants' educational background, gender, and prior programming experience. The questionnaire aimed to gather comprehensive information from participants before they engaged with the low-code platform. Participants were asked to provide details regarding their demographic background, including gender identity, academic qualifications, age, and current country of residence. Their level of experience in programming or software development, knowledge of databases, and familiarity with the concept of low code were assessed. Additionally, participants rated their confidence in explaining low-code development to others and expressed their initial attitudes toward low-code development. Opinions on whether low-code development poses a challenge for those without an IT background, its impact on the speed of application development, and its perceived difficulty were also captured. This comprehensive set of questions aimed to establish a baseline understanding of participants' backgrounds, experiences, and attitudes before their involvement in the low-code development experiment.

5.4 Interaction with the Low Code Platform

Students were given instructions on setting up Mendix Studio Pro. Additionally, they received a task document that required completion and submission upon finishing. The initiation of the experiment began when students successfully downloaded Mendix Studio Pro and opened the project. The experiment included three distinct tasks, carefully selected to represent examples that could be encountered in a software development project. These tasks were designed to evaluate students' interaction with the Mendix low-code platform across various dimensions, including performance, experience, and attitude toward low-code development. Each task was structured to present a unique set of challenges and opportunities inherent in utilizing low-code solutions and the relevance of the tasks to typical real-world scenarios they might encounter as developers. The tasks are described below in detail:

- Task 1: Entity Creation: In the assigned task for students within the low-code platform experimentation, the initial focus was on entity creation within the MovieManagement module. Students were instructed to navigate through the domain model and undertake the following steps:

first, create a new entity named ‘Actor’ with specific attributes, including FullName (String), Email (String), and DateOfBirth (DateTime). The subsequent step involved establishing a one-to-many association between the newly formed ‘Actor’ entity and the pre-existing ‘Role’ entity. To validate their execution, students were asked to provide a screenshot depicting the completed task. Furthermore, to assess their understanding and application of the task, students were required to describe the schema of the resulting database. This entailed articulating observations related to entities and associations, thereby providing insight into the structural elements of the database resulting from their entity creation. The difficulty of this task was measured on a scale from 1 to 5, allowing students to subjectively rate the complexity they encountered during the completion of the entity creation process. In the comments section, students were encouraged to share their reflections on the task. This open-ended feedback provided an opportunity for students to communicate any challenges faced, articulate thoughts on the overall experience, and offer valuable insights into their learning journey within the low-code platform.

- Task 2: Pages and Microflows: In Task 2, participants engaged with the Mendix low-code platform, focusing on Pages and Microflows. The sequence unfolded with the directive to access the “Actor_Overview” page, where participants were then prompted to integrate a "Data Grid" widget, specifying the data source as a database. Subsequently, participants were required to provide a screenshot of the "Actor_Overview" page, showcasing the integrated "Data Grid" displaying actor information. Additionally, students were tasked with describing the ‘Movie_NewEdit’ page (Figure 1), outlining its elements, purpose, and potential user interactions. Participants had to determine whether they successfully linked the ‘Save’ button on the ‘Movie_NewEdit’ page to the ‘ACT_Movie_Save’ Microflow (Figure 2), followed by an explanation of the microflow’s functionality. To capture the perceived difficulty of the task, participants rated it on a scale from 1 to 5, with 1 being very easy and 5 being very difficult. The open-ended comments section allowed participants to share reflections, challenges faced, and additional insights into their experience with this practical aspect of the Mendix platform. Through Task 2, the study aimed to assess participants’ hands-on proficiency and their ability to effectively navigate and utilize key components of the low-code environment.
- Task 3: Database Enhancement: The third task initiated with participants accessing the Domain Model. Their primary objective centered around optimizing the database structure to effectively handle ticket reservations. Guided by a provided ticket (Figure 3), participants were instructed to ensure that the database comprehensively captures all pertinent information for a resilient reservation system. This optimization necessitated the introduction of additional attributes to existing entities, facilitating any required modifications to tailor the database for an efficient reservation system attuned to the specifics of the provided ticket.

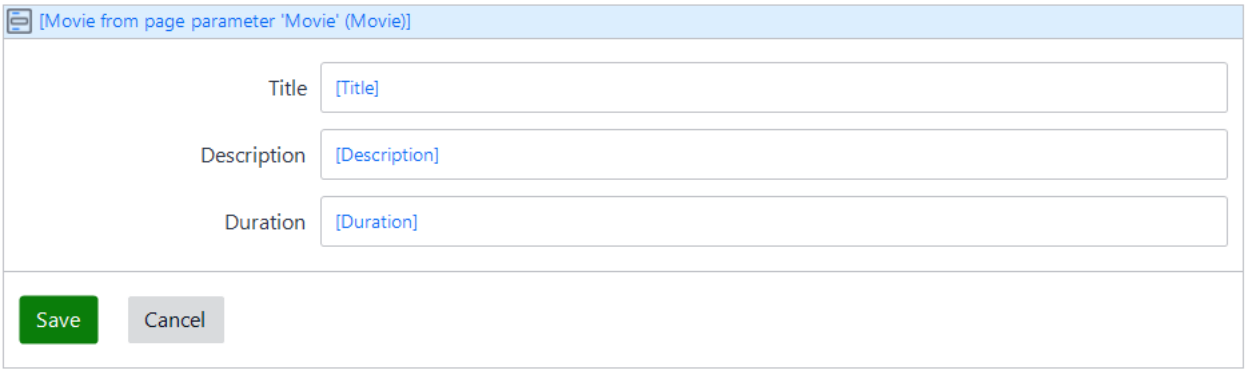

Figure 1. ‘Movie_NewEdit’

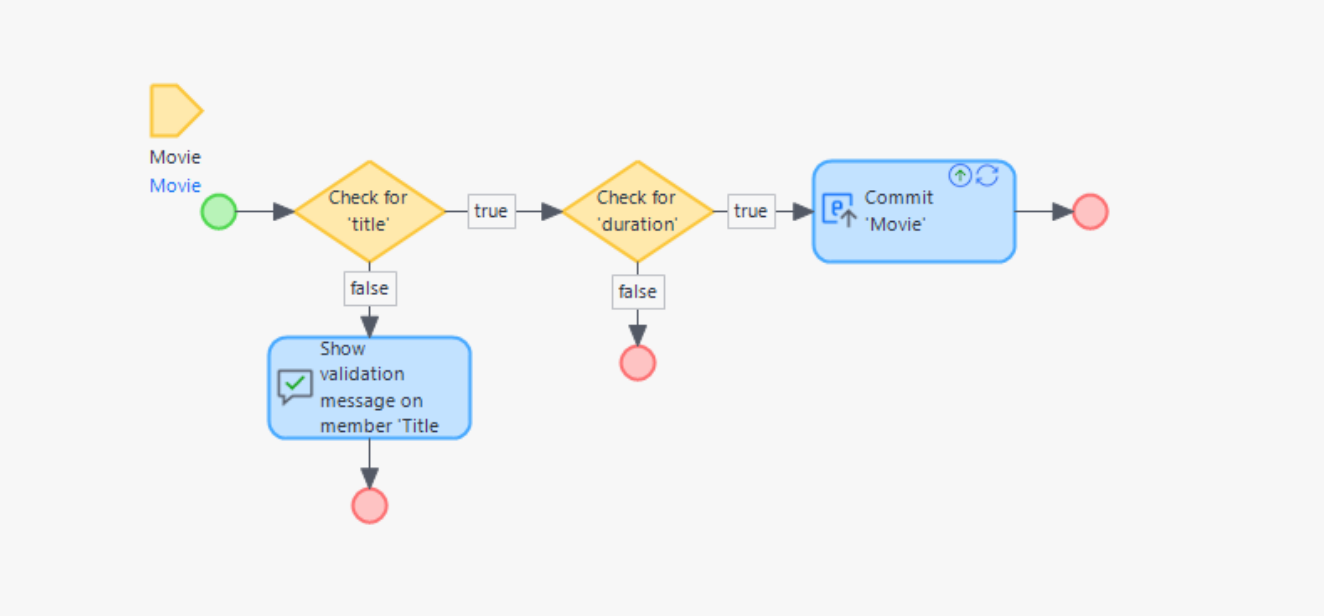

Figure 2. Microflow: ACT_Movie_Save

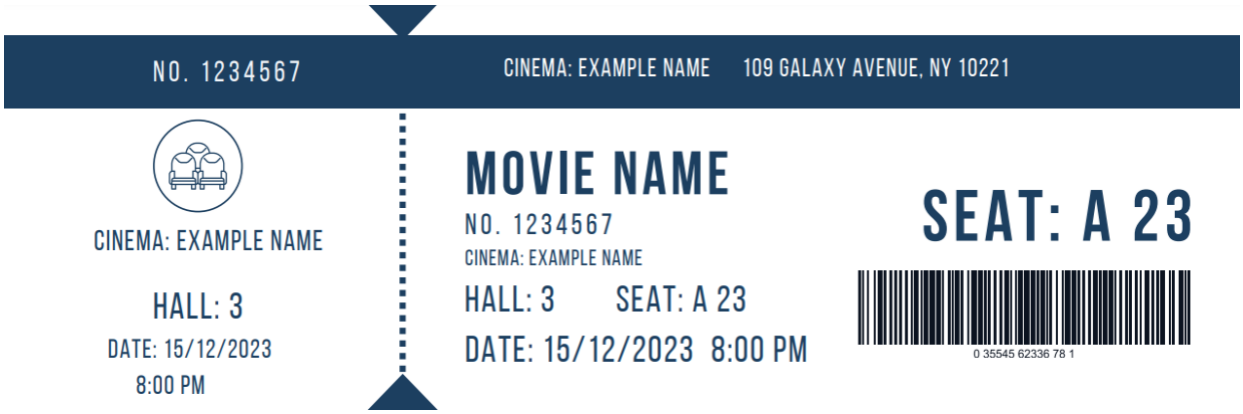

Figure 3. Ticket

As a tangible representation of their work, participants were required to add a screenshot in the document with a visual depiction of the enhanced database structure resulting from their optimizations. To gauge the perceived difficulty of the task, participants were asked to self-assess on a scale from 1 to 5, where 1 indicated very easy and 5 denoted very difficult. The open-ended comments section provided a qualitative dimension to participants' experiences, enabling them to share challenges faced, notable observations, and reflections on the database enhancement process within the Mendix low-code environment. Through Task 3, the study aimed to evaluate participants' proficiency in database optimization and their ability to tailor the database structure for real-world applications, such as ticket reservations. Following the completion of the tasks, participants were instructed to send their responses back to the researchers and fill in the final questionnaire. The final questionnaire sought participants' perspectives on various aspects related to low-code development, encompassing their attitude towards it, their experience with the low-code platform employed in the study, and their perception of the impact of low-code development on the speed of application development. Additionally, participants were prompted to provide descriptive insights into their overall experience, offering reflections on their engagement with the low-code tasks and sharing their thoughts on the broader implications of low-code development. The final questionnaire sought to gather detailed feedback, illuminating participants' attitudes, experiences, and reflections.

5.5 Post Questionnaire

The concluding step, following the submission of the deliverable containing their responses, involved completing the final questionnaire. In the post-test questionnaire, participants were invited to provide reflections on their experience with the low-code platform, offering insights into their evolving perceptions and capabilities. They reevaluated their attitude towards low-code development, categorizing it as "Very negative," "Somewhat negative," "Somewhat positive," or "Very positive." Participants then characterized their experience with the low-code platform, choosing from descriptors such as "Limited," "Challenging," "Positive," and "Exceptional." Additionally, participants assessed the platform's impact on the speed of application development, rating it on a scale from "Does not improve speed at all" to "Significantly improves speed." Finally, participants reconsidered the perceived difficulty of low-code development, categorizing it as "Very Challenging," "Challenging," "Easy," or "Very Easy." These post-test questions aimed to capture any shifts in participants' attitudes, experiences, and perceptions following their hands-on interaction with the low-code platform.

6. Results

6.1 Initial Attitudes Towards Low-Code Development and Demographic Factors

Participants were asked about their attitudes towards low-code development before taking part in the experiment. The responses reflected a predominantly positive sentiment, with most participants indicating a "somewhat positive" attitude. Out of the total respondents, 27 participants expressed a positive disposition, while only 1 participant reported a "somewhat negative" attitude. This overall positive inclination suggests that, prior to the experiment, participants generally held favorable views toward low-code development. In light of the non-normal distribution of the data, as confirmed by the Shapiro-Wilk test ($p < 0.001$), an analysis using the Mann-Whitney test was employed to address RQ1.

This analysis aimed to explore potential differences in initial attitudes toward the low-code platform considering demographic factors, namely gender, country, and IT background. The results revealed no significant country-based correlations ($U = 91$, $p = 0.691$), indicating that initial attitudes did not vary significantly across different countries. Similarly, no statistically significant correlations were observed with IT background ($U = 51$, $p = 0.299$), suggesting that individuals with varying IT backgrounds did not exhibit distinct initial attitudes. Moreover, gender correlations did not reach statistical significance ($U = 72$, $p = 0.211$), indicating that the initial attitudes were not significantly influenced by gender.

6.2 Low Code Experience and Demographic Factors

Following their participation in the experiment, participants assessed their experiences with the low-code platform using a Likert scale ranging from 1 to 4. The majority of participants (23) reported a positive experience, expressing satisfaction or comfort with the low-code platform. However, a subset of participants (4) found the experience challenging, suggesting some difficulties or complexities encountered during their interaction with the platform. One participant used the term "exceptional" to describe their experience, implying an exceptionally positive and favorable encounter with the low-code platform.

To assess the normality of the data, a Shapiro-Wilk test was conducted, revealing a non-normal distribution ($p < 0.001$). Subsequently, the Mann-Whitney test was employed to address RQ2. The results indicated a statistically significant difference in experience ratings between male and female participants, with the male group showing significantly higher ratings ($U = 56$, $p = 0.008$). Additional analyses explored differences between participants' experience ratings and demographic factors, including IT background ($p < 0.701$) and current country of residence ($p > 0.178$). However, no statistically significant differences were identified for these demographic variables. Consequently, the findings of this study suggest that, aside from the observed gender differences, participants' experiences with the low-code platform were not influenced by their academic qualifications, age, or country of residence, as no statistically significant differences were detected in relation to these demographic factors.

6.3 Perception of the difficulty of low-code development

To address RQ3, the changes in participants' perceptions of low-code development difficulty before and after using the low-code platform were analyzed. Initially (Figure 4), perceptions were mixed: one participant found it 'very easy,' 15 'easy,' and 12 'challenging.' Post-interaction Figure 5, a positive shift was observed: the number of participants who found it 'very easy' increased to 4, while those perceiving it as 'easy' remained at 15. Those who considered it 'challenging' decreased to 9.

This indicates that for 10 participants, the platform's use clarified its ease, shifting their perception positively. Notably, 18 participants maintained their initial perception, suggesting a consistent experience with the platform's usability. The absence of any negative shifts implies that the interaction was, at the least, not detrimental and may hint at an intuitive design. The data suggests that while some aspects of the platform may still pose challenges, overall, the interaction had a favorable influence on user perceptions of low-code development.

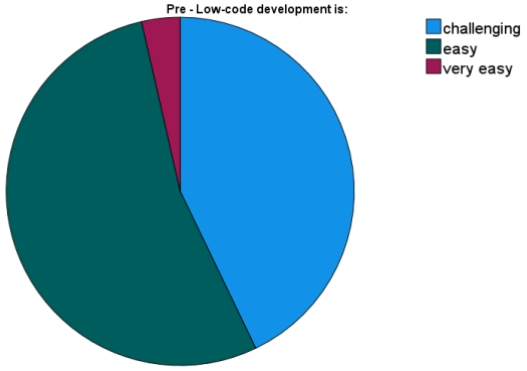

Figure 4. Pre - Low code development is

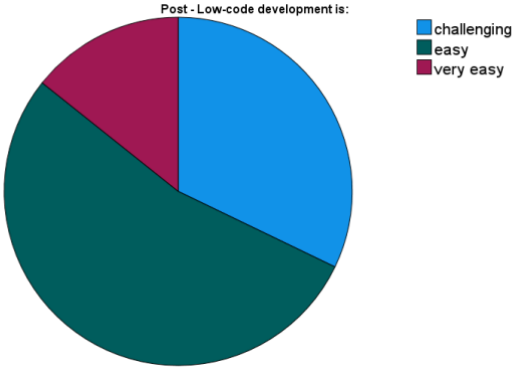

Figure 5. Post - Low code development is

6.4 Performance

Based on the received deliverables, the evaluation process involved two reviewers for each task, assigning a score ranging from 0 to 100 points. The results are shown in Table 1. The first task required students to present a screenshot displaying the association and attributes while explaining the domain model. A full 30 points were awarded if the screenshot accurately depicted the relation, and the explanation was deemed correct. In the second task, students were instructed to connect the datagrid with the page, elucidate the page's content, and link a button with the microflow. Additionally, they were tasked with describing an already existing microflow and explaining its functionality. Here, a perfect execution earned them 40 points. Screenshots played a crucial role in all tasks, ensuring that every detail was appropriately captured. The final task involved creating a database structure in Mendix Studio Pro for cinema ticket management. Students were tasked with designing a database ticket to handle ticket reservations. A comprehensive screenshot demonstrating the correct attributes and a precise explanation warranted the full 30 points. Through this structured evaluation approach, the aim was to thoroughly assess each student's comprehension and execution of the given tasks in Mendix Studio Pro. Figure 6 serves as an illustrative example of a response received for Task 3.

Table 1. Descriptive Statistics

	N	**Minimum**	**Maximum**	**Mean**	**Std Deviation**
Performance	28	70.00	100.00	92.8571	8

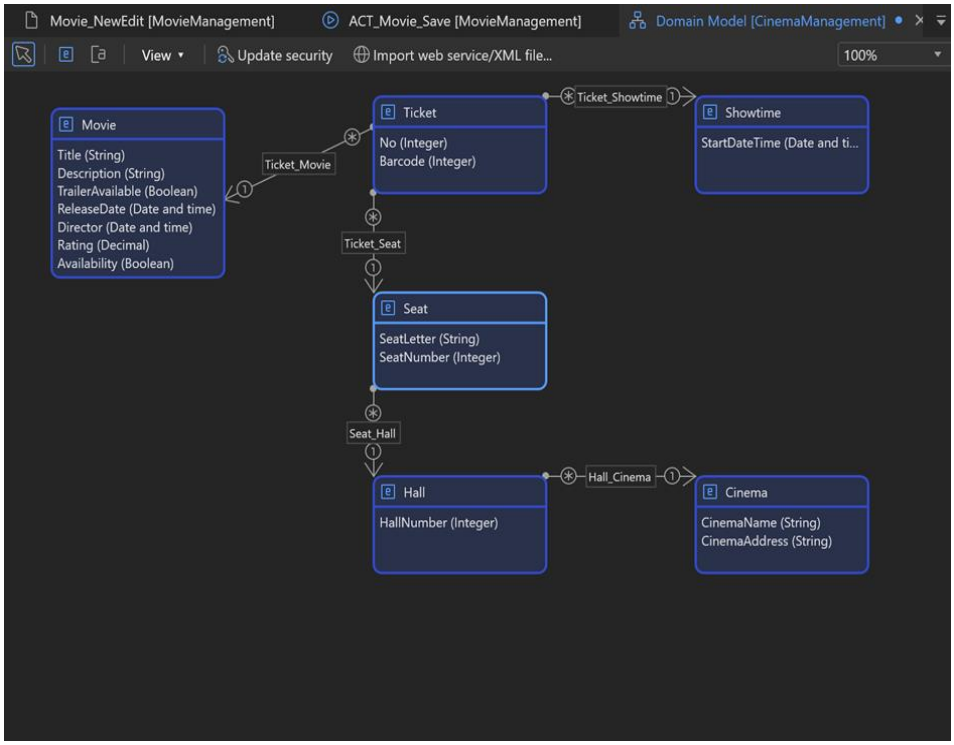

Figure 6. Example of a response of a student in the third task

To address RQ4, an examination of performance score differences between participants with an IT background and those without was undertaken. A Mann-Whitney U test was conducted, revealing a statistically significant discrepancy in performance scores ($U = 24{,}000$, $p = 0.01$). This statistical significance emphasizes that an IT background is correlated with variations in performance within the context of this experiment.

6.5 Insights from Task Deliverables

An analysis of student deliverables revealed a significant comprehension gap in certain concepts within the low-code platform. While some initially struggled with articulating associations in the database concept, a clearer understanding emerged in the final task, especially concerning entity and attribute differentiation. In contrast, participants with an IT background exhibited proficiency, showcasing a robust understanding during design and creation. The disparity in terminology and comprehension played a pivotal role, with those familiar with IT concepts demonstrating a more adept grasp. Although all students completed tasks successfully, distinctions lay in response precision and overall comprehension. Diverse participant reflections offer insights into low-code platform experiences. Some faced challenges initially but found promise as usage became clear. Participants without prior low-code knowledge acknowledged challenges but recognized the platform's potential, especially for non-programmers.

Feedback highlighted the learning curve, with some revisiting instructional videos. Despite varied experiences, overall sentiment towards the low-code platform was positive, appreciating its capacity to simplify code development and streamline database creation. Those with prior low-code experience emphasized its straightforward nature, emphasizing the impact of familiarity on performance and attitude. The study contributes insights into the dual nature of low-code platforms, offering a comprehensive view of future research and development in this domain. The observed differences in performance between students with IT backgrounds and those without, as reported in this experiment and supported by literature [13] point to the need for tailored educational strategies. In particular, integrating low-code platforms into the curriculum can serve as a practical tool for teaching software development. This approach allows all students to get involved in software creation through direct experience.

This study outlines essential elements in learning low-code, such as the query phase, preprocessing, segmentation, feature extraction, and matching strategies. These can guide the structuring of courses, combining theory with hands-on practices. By doing so, the educational content will not only meet current industry demands but also cater to the varied expertise of students, paving the way for preparing skilled professionals in software development.

It is also suggested that curriculum planners should include modern tech concepts in their courses. By doing so based on these results, it can be ensured that educational programs keep pace while being accessible to a diverse student body. This approach is vital for creating an equitable learning space that can produce the tech innovators and leaders of tomorrow.

7. Limitations

Although this research is making an important contribution to the current state of research, it also has its limitations. The data acquired through the questionnaire are based on subjectivity or human understanding. This lack of objectivity can add some prejudice or bias to the outcome. Additionally, it cannot be assured that the set of potential factors or decision variables used in the research was exhaustive.

Although their identification was premised on the extensive literature review still the possibility of other significant factors cannot be ignored. In this context, the research may not fully encapsulate the breadth of variables affecting low-code platform efficacy, such as long-term retention of skills. Therefore, a deeper study in the future can provide a more consistent analysis of the current scenario. In terms of improving the ability to generalize and the robustness of the results, a quantitative empirical study incorporating a larger sample can be the direction for future research. Additionally, the findings are based on a specific demographic, which may not be representative of the wider population engaging with low-code platforms, thus potentially limiting the generalizability of the results.

8. Conclusion

In conclusion, this work examines students' experiences and perceptions of a low-code platform, with a specific focus on exploring the factors influencing low-code performance, experience, and attitude. The research, guided by four distinct RQs, investigates various dimensions of participants' engagement with the Mendix low-code platform. Notably, three out of the four RQs center around perceptions, highlighting the importance of understanding user sentiments in the adoption of low-code platforms, while the fourth RQ explores tangible performance metrics.

The analysis of initial attitudes towards low-code development, as scrutinized in RQ1, revealed an overall positive inclination among participants. However, no significant differences were observed based on demographic factors such as gender, country, or IT background, underlining a consistent positive sentiment across diverse groups. Moving on to RQ2, which focused on participants' experiences with the low-code platform, statistical differences were identified between male and female participants, with males reporting higher satisfaction. However, no significant variations were found based on academic qualifications or current country of residence. RQ3 investigated participants' perceptions of the difficulty of low-code development before and after interaction with the platform. While there was an overall positive shift in perceptions post-interaction, some participants still found certain aspects challenging.

The final Research Question (RQ4) revealed a significant correlation between an IT background and performance scores, highlighting the crucial role of coding knowledge in low-code proficiency. While all participants showed commendable performance, those with an IT background demonstrated a deeper comprehension and specificity in their responses. The analysis of student deliverables exposed proficiency disparities, emphasizing the platform's positive reception and potential for simplifying code development and database creation.

In summary, this work contributes valuable insights into the multifaceted aspects of low-code platform adoption within an educational context. While perceptions and experiences play a crucial role, the study underscores the pivotal importance of tangible performance metrics, especially in correlation with participants' coding backgrounds. These findings provide a foundation for future research and development in the dynamic landscape of low-code platforms, catering to diverse user-profiles and enhancing the overall usability and effectiveness of such innovative tools.

This work was supported by the School of Engineering and Digital Sciences, Nazarbayev University. (Corresponding author: Mariza Tsakalerou.)

References

[1] Magic Quadrant for Enterprise Low-Code Application Platforms," Gartner, [Online]. Available: Magic Quadrant for Enterprise Low-Code Application Platforms. [Accessed: February 7, 2024].

[2] Sahay, A., Indamutsa, A., Di Ruscio, D., and Pierantonio, A. (2020, August). Supporting the understanding and comparison of low-code development platforms. In 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (pp. 171-178). IEEE.
[3] Paul Vincent, Kimihiko Iijima, Mark Driver, Jason Wong, and Yefim Natis. Magic quadrant for enterprise low-code application platforms. Gartner report, 2019
[4] Çakıroğlu, Ü., Çevik, İ., Köşeli, E., & Aydın, M. (2021). Understanding students' abstractions in block-based programming environments: A performance based evaluation. Thinking Skills and Creativity, 41, 100888.
[5] M. L. How, S. M. Cheah, Y. J. Chan, A. C. Khor, and E. M. P. Say, "Artificial Intelligence for Advancing Sustainable Development Goals (SDGs): An Inclusive Democratized Low-Code Approach," in The Ethics of Artificial Intelligence for the Sustainable Development Goals, pp. 145-165, Cham, Springer International Publishing, 2023.
[6] J. Metrôlho, F. Ribeiro, P. Graça, A. Mourato, D. Figueiredo, and H. Vilarinho, "Aligning Software Engineering Teaching Strategies and Practices with Industrial Needs," Computation, vol. 10, p. 129, 2022.
[7] Avishahar-Zeira and D. H. Lorenz, "Could No-Code Be Code? Toward a No-Code Programming Language for Citizen Developers," in Proceedings of the 2023 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software, pp. 103-119, October 2023.
[8] E. Martinez and L. Pfister, "Benefits and limitations of using low-code development to support digitalization in the construction industry," Automation in Construction, vol. 152, p. 104909, 2023.
[9] R. Picek, "Low-code/No-code Platforms and Modern ERP Systems," in 2023 International Conference on Information Management (ICIM), March 2023, pp. 44-49.
[10] L. Östlund, N. Wicklund, and R. Glassey, "It's Never Too Early to Learn About Code Quality: A Longitudinal Study of Code Quality in First-year Computer Science Students," in Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1, March 2023, pp. 792-798.
[11] J. Luo and M. Hou, "Innovation of the Higher Education Grassroots Statistical Reports System based on Low-Code Development," in 2023 7th International Conference on Management Engineering, Software Engineering and Service Sciences (ICMSS), January 2023, pp. 36-40.
[12] Lang, D., Handley, M., Erdman, A. M., Park, J. J., and Tsakalerou, M. (2019, June). "Intercultural competency differences between US and Central Asian students in an engineering across cultures and nations graduate course." In 2019 ASEE Annual Conference & Exposition.
[13] Cao, Q., Seow, C. K., Lim, L. H. I., Keoh, S. L., Dale, V., Honeychurch, S., Tasler, N. and Bremner, D. (2023). Learners' differences in blended learner-centric approach for a common programming subject. International Journal of Information and Education Technology, 13(6), 906-913.