

Article

Measuring Students' Satisfaction on an XAI-Based Mixed Initiative Tutoring System for Database Design

S. M. F. D. Syed Mustapha 

College of Technological Innovation, Zayed University, Dubai P.O. Box 19282, United Arab Emirates;
syed.duani@zu.ac.ae

Abstract

This research proposes the development of an Entity-Relationship Diagram—PRO (ERD-PRO) to assist students in understanding the concept of developing Entity-Relationship Diagrams in designing a database. ERD-PRO is an Intelligent Tutoring System (ITS) that is built using a mixed-initiative approach to address the learning challenges by adopting Explainable Artificial Intelligence (XAI) concept to provide individualized and on-demand feedback and guidance. The effectiveness of ERD-PRO is tested on 25 participants from different educational institutions. Pre- development surveys are conducted to determine learning needs and post-development surveys are performed to measure the success. The results show that the design of ERD-PRO, guided by survey findings, successfully addresses key challenges in database design education. 65% of students agreed that the system's explanation facilities effectively clarified difficult topics, and 90% expressed high satisfaction with the tool. The integration of XAI features within ERD-PRO has enhanced its ability to provide meaningful, scenario-based explanations, demonstrating its potential as an effective intelligent tutoring system. These findings validate the effectiveness of ERD-PRO in meeting its objectives and highlight its value in providing tailored explanations for database design instruction.



Academic Editors: Teen-Hang Meen
and Chun-Yen Chang

Received: 19 June 2025

Revised: 18 October 2025

Accepted: 24 November 2025

Published: 2 December 2025

Citation: Mustapha, S.M.F.D.S. Measuring Students' Satisfaction on an XAI-Based Mixed Initiative Tutoring System for Database Design. *Appl. Syst. Innov.* **2025**, *8*, 189. <https://doi.org/10.3390/asi8060189>

Copyright: © 2025 by the author. Published by MDPI on behalf of the International Institute of Knowledge Innovation and Invention. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: intelligent tutoring systems; database design; entity-relationship diagrams; explainable AI; educational technology

1. Introduction

The Intelligent Tutoring Systems (ITSs) are designed to provide personalized instruction and feedback to learners, making education more effective and engaging [1]. Implementing ITS for teaching databases had its significance to support students in integrating knowledge about database concepts, building SQL codes, designing ERD and performing table normalization [2–4]. According to Freeman & Zachar [5], a traditional ITS typically comprises four major components including expert module, pedagogical module, student module and user-interface module.

Active learning strategies, which directly engage the learner in the process of learning, have been shown to improve knowledge retention and skill acquisition. Assessment strategies, including pre- and post-tests, provide feedback on student learning and show where further support is necessary [6]. Incorporation of such learning principles into intelligent tutoring systems can be used to enhance one-on-one learning and assure instructional interventions are effective and evidence-based [7].

An Entity-Relationship Diagram (ERD) is a pivotal component of the database schema [8]. It is a visual model used widely to illustrate business entities, their attributes,

and the relationship between them [1]. Sketching an ER model is the first step in the database design process, which creates a logical Entity Relationship Diagram (ERD). It is then transformed into a relational model followed by well-defined rules to be translated into a physical database and aligned with a selected software platform for effective implementation [9]. Considering this as the essential step, it is our motivation to enhance the learners' learning experience in ERD design by developing ERD-PRO, a tutoring system that provides explanation and reasoning support using a mixed-initiative approach as its learning pedagogy [10,11].

A single mistake in designing an Entity-Relationship Diagram can cause significant problems in the database [12]. The complexity of concepts and design rules poses a risk that students may carry alternative views into their professional careers [13]. Consequently, the outcome could lead to an erroneous data model. Rashkovits and Lavy [14] stated that novice students correctly identify entities and attributes. However, they encounter significant challenges in learning the connectivity of relationships and cardinality between entities. Their research proposes to assist students with consultation tools to help construct ERD as these students face issues of cognitive complexity, therefore allowing errors to occur. The lack of immediate feedback and information overload creates confusion, especially in data models crafted by students in the early stages of database learning, where errors and inaccuracies may arise due to the complexity of the material and the way it is taught [15]. Alongside novice students, students of advanced database studies can sometimes create suboptimal database designs due to the complexity of elements [16]. Such will require them to assimilate and practice the design elements of ERD, thereby making the integration of an ITS inevitable [14,17]. The implementation of Artificial Intelligence with ITS also promises to enhance learning for the students [1], in a way that mirrors the rigor required in designing accurate Entity Relationship Diagrams.

Research Questions

The present study aims to investigate the key challenges students face in designing ERDs, and to develop a tutoring system (ERD-PRO) that addresses these challenges. The resulting ERD-PRO uses a mixed-initiative approach by providing two complementary components: an explanation module and a reasoning module, which together support students' understanding of ERD design. The research poses the following research questions, which need to be addressed:

1. What are the challenges faced by students in designing ERDs?
How can ERD-PRO address those challenges?
2. Will the mixed-initiative approach improve the explanation and reasoning modules in ERD-PRO?
3. What are the perceptions of students about ERD-PRO's features related to explanation and reasoning?

The research explored the challenges of database design by focusing on entity, attribute, and relationship identification using ERD. It aims to improve the design of databases by analyzing the approaches of beginners and advanced students and helping them expand their use of ERD-PRO, along with offering insights to advance education and to guide best practices for practitioners.

The following sections describe a literature review on Intelligent Tutoring Systems, the concept of XAI, and how its integration with the ITS concept, research gap, the development of ERD-PRO, its workflow and framework, and finally the research output and analysis.

2. Literature Review

2.1. Intelligent Tutoring System for Database

Intelligent Tutoring System (ITS) enhances personalized learning using AI [18]. Advancements in ITS improve specific subjects significantly, providing one-on-one teaching assistance, guiding problem-solving [19], and offering prompt feedback. It fosters intelligent learning environments [5], reducing teacher intervention [1]. A typical ITS includes a student, tutor, and domain model, with a user interface for communication [20]. The learner model identifies student nuances to provide precise responses, the tutor model employs advanced strategies to guide, and the domain model offers educational content. The goal is to effectively acquire domain-specific knowledge. ITS is part of AI in Education (AIED) [21], emphasizing personalized instructions using interactive methods [22]. Studies show ITS effectiveness over traditional teaching, especially in STEM [19,23].

Intelligent tutoring programs that include AI-enabled functions and capabilities in problem-solving enhance interactive learning engagement [19]. They differ from traditional computer-based instructions, which lack the problem-solving capabilities essential for aligning with pedagogical theories [20]. Modern ITS focuses on active learning theories, such as situated learning and cognitive processes [22], emphasizing student engagement over passive instruction [18]. ITS delivers tailored, immediate instructions or feedback [10], using cognitive science and AI to aid skill acquisition [24]. They emulate human tutors by adapting methods to student abilities, supporting individual adaptation in educational processes [22].

Despite its efficiency, some research indicates that this approach leads to shallow knowledge and difficulty in applying it to new challenges [20,22]. Metacognitive activities like self-explanation and reflection can overcome these issues [25]. Self-explanation helps individuals clarify new information [5], refining knowledge for future use. Tutoring dialogues, supporting self-explanation and reflection, are used in many ITS for deeper learning [24].

ITS development faces challenges in accurately portraying and assessing a student's cognitive state and requirements amid imprecise data, deploying AI techniques like Bayesian networks, neural networks, fuzzy logic, and rule-based systems are necessary [18,21]. AI technologies, such as natural language processing and machine learning, have been employed to understand students' learning difficulties and provide personalized recommendations. XAI programs aim to make machine learning processes explainable and comprehensible, akin to ERD modeling explanations [20].

2.2. Explainable AI (XAI)

XAI, or Explainable AI, aims to make machine learning processes transparent and understandable, which is particularly critical in finance and healthcare [26]. This approach addresses the drawbacks of traditional AI by emphasizing accountability, traceability, and interoperability [21,27]. Interpreting and explaining machine learning algorithms to improve them and ensure accountability is essential [17]. XAI is prevalent in deep learning libraries like Google Collab, with frameworks focusing on interpretability criteria such as usability, causality, and reliability, shaping the field [23,27,28].

The goal of XAI is to empower end users to trust, understand, and manage advanced AI systems by developing explainable machine learning techniques through effective explanation methods [26]. This fosters greater reliance on AI decisions, actions, and recommendations, achievable only with a clear understanding of the system's rationale [27]. XAI also emphasizes the autonomy of decision policies in autonomous systems, data analytics, and multimedia event classification, resolving issues in reinforcement and supervised learning [23].

Lamy et al. [29] describe how intelligent systems, created with the aid of XAI, explain their recommendations by interpreting predictions for less understandable algorithms and interpretable models, such as rule-based ones. Visualization, presenting information in a visual format, is another effective approach [27]. XAI in ITS for database design is implemented differently, emphasizing detailed step-by-step explanations to enhance students' understanding of Entity-Relationship Diagram (ERD) development. This approach ensures that each step in the design process is clearly demonstrated, providing students with insights into the reasoning behind each decision. By making the underlying logic of ERD development transparent, students can better understand the complexities involved, thereby improving their ability to accurately identify entities, attributes, and relationships. This method not only clarifies complex concepts but also builds students' confidence and competence in applying these skills to real-world scenarios.

2.3. XAI and ITS

Adopting XAI in ITS is essential in various ways. XAI addresses challenges related to learner agency and noisy learning data [21], driven by stakeholder requirements for explanations in education [30]. Educators need explanations to diagnose class focus insights, provide individual feedback, ensure accountability, and consult parents. Student feedback helps improve learning, motivation, and self-esteem [31]. Teacher feedback evaluates teaching effectiveness, learning design, and individual student needs [32]. Data-backed explanations support decision-making, improve efforts, and develop institutional profiles aiding administrators [33].

In military training, XAI asks users about entities and events in past simulations [5]. Effective XAI systems prompt users to ask questions and understand answers, improving based on past interactions [20]. Personalized learning benefits from understanding the “why” of predictions for tailored support [21]. Educational software explains the “what” and “why” to learners, building trust and usefulness [30].

XAI in education combines artificial intelligence, Human-Centered AI, and Human-Computer Interaction. The XAI-ED framework includes six factors: human-centered design, model classes, explainable presentation approaches, roles, stakeholders, and potential drawbacks [21,30]. ITSs usually avoid detailed XAI techniques but aim to be helpful and explain generally [31]. The role of AI and XAI in evolving ITSs in education and training is crucial, emphasizing self-monitoring and reflection [21–23,25]. AI developments form the basis for smarter ITS, focusing on improving explainable model predictions [33]. Early XAI explanations were static; recent research shows the need for interaction, context, task, and user consideration [25]. Maintaining communication between users and AI requires a shared language, such as NLP, simulations, and graphics [5].

The concept of XAI in ITS extends beyond technical interpretability to traditional pedagogical transparency, such as Open Learner Models (OLMs), which display the system's assessment of a student's knowledge and abilities. However, the concept of XAI is to embed explanations directly into the adaptive hinting mechanism of the ITS. Our work has adapted this concept from the work by Conati et al. [27] on the Adaptive CSP or ACSP applet for learning constraint satisfaction problems in which the explanations were designed around two dimensions:

- Why explanations: describing the reasons a particular hint was given (e.g., why the student was classified as a “lower learner” at that point).
- How explanations: describing the processes to generate the hint (e.g., how the system scored user behavior and ranked possible hints).

These explanations improve not only student comprehension of the domain but also their trust and acceptance of the tutor's guidance, while highlighting the need for

personalization because not all students benefit equally from the same type of explanation. Students could click an “Explain Hint” button to receive layered explanations. For example:

- Why am I delivered this hint? → explained the user’s classification and related behaviors.
- Why am I predicted to be lower learning? → linked the student’s actions to specific rules and weights.
- How was my hint chosen? → showed the ranking process among candidate hints.

Based on the conceptual approach of XAI by Conati et al. [27], the approach taken by ERD-PRO is to provide two levels of hint which are Reasons and Explanations. The Reasons justify the choices suggested by the system in the context of the scenario while the Explanations provide broader context of the choices. For example, the car plate number may be use characters as the data type since the context of the scenario used alphanumeric (Reasons), however, in broader sense the car plate number may use both numeric or character data type. These two hints are separated with the purpose of recognizing the two levels of conceptual understanding of the student. The demonstration of XAI-based ITS on ERD-PRO is described in ERD-PRO section.

2.4. Research Gap

Although XAI and ITS witness significant advancements but various challenges still hinder personalized delivery of content devoid of meaningful and adaptive explanations [34]. For instance, Melo et al. [31] mention that most XAI systems rely on a rigid question-answer paradigm, which lacks meaningful information exchange. While Clancy and Hoffman [20] stated that explanations are often generic, they cannot therefore adapt to the diverse understanding capacities of learners. In short, existing models struggle with explaining complex processes, especially those related to visual content. Sharma [35] emphasized that effective ITS systems should use instructional principles by promoting self-explanations to develop deep learning. Elmadani et al. [24] emphasize the use of ITS logs and analysis of students’ behavior to support timely interventions and error predictions. This research gap presents opportunities for ERD-PRO to provide tailored explanations, such as user-adapted and scenario-specific explanations, thereby enhancing relevance and engagement. Collaborative content creation integrated with crowd sourcing to improve the quality and diversity of explanations and granular learning. This approach offers explanations with different levels of depth suited to individual needs.

3. Research Methodology

The study employed a mixed-methods research design to evaluate the efficacy of ERD-PRO as an Intelligent Tutoring System (ITS) for database instruction. The research design consisted of two phases: a pre-development survey, which aimed to identify the challenges faced by students in ER diagram design, and a post-development survey that quantitatively assessed the extent to which ERD-PRO addressed these challenges (kindly refer to Appendices A and B, respectively). The combination of qualitative and quantitative methods was useful to realize the complicated character of learning because it allowed for both careful exploration of students’ difficulties and measurable evidence of improvement [36].

3.1. Theoretical or Conceptual Framework

Three complementary conceptual frameworks were backing the research. First, the use of a mixed-methods design permitted a deeper and more balanced understanding of the problem [37], combining the exploratory capability of qualitative inquiry with the systematic vigor of quantitative measurement [38]. Second, Constructive Learning

Theory [39] informed ERD-PRO's design by emphasizing engaged participation in the construction of knowledge. The system was designed to enable students to repeat and replicate, experiment with solutions, and receive immediate feedback and guidance, all of which facilitate the process of creating a deeper understanding. Bloom's Taxonomy also offered a model for assessing learning outcomes [40], namely the progression from low-order recall of ERD material to higher-order skills, such as analysis, application, and evaluation. Both perspectives combined establish the tone for ERD-PRO design and its assessment here.

3.2. Participant's Sampling

Participants were sampled for a specific purpose with the support of a homogeneous method to enable the sharing of relevant academic characteristics [38,39]. The homogeneous technique was applied to ensure that all participants shared the relevant academic characteristics required for meaningful comparison, complementing the purposive sampling criteria and providing a more focused sample of students with comparable knowledge of ER diagrams. Students were qualified if they had a CGPA of over 3.0, which reflects good academic standing and is roughly equivalent to a B or higher in the US 4.0 GPA system [41]. Advanced learners were also expected to have a minimum grade of B+ in an introductory database course. Recruitment was conducted with the agreement of two universities, one in the United Arab Emirates and the other in Malaysia, and participation was entirely voluntary. These requirements ensured that respondents had sufficient familiarity with ER diagrams to provide informed opinions and accurately represent the targeted user group.

Because of these limitations, the sample population was necessarily small, consistent with current research that employs purposive sampling in expert environments [38]. Twelve students volunteered to take part. To balance for variation in prior knowledge, the sample was stratified into novice and advanced learners: the latter were taking first-level database courses like CS266, CS264, CS240, CS255, and CS230, whereas the former were taking advanced-level database subjects. This division was required for capturing variation in perceptions among individuals with minimal knowledge and those possessing more expertise [39].

3.3. Data Collection

The pre-development survey was conducted with the objective of determining the key problems students faced in ERD design. It focused on areas such as identifying relevant entities and attributes, defining relationships, applying cardinalities, and applying business rules. Semi-structured interviews were conducted with upper-level students from both institutions, as the students were already familiar with ERD concepts and could therefore comment effectively on areas for improvement. The interview guide was piloted with faculty experts to ensure relevance and clarity, and with a small group of students prior to the main study. Ethical practices were maintained throughout, including obtaining informed consent, ensuring voluntary participation, and maintaining the confidentiality of responses.

The post-development survey was administered after students had utilized ERD-PRO. To ensure appropriate exposure, each student was required to use the system to solve at least ten practice cases before completing the survey. The instrument was developed primarily as an official quantitative survey, where closed-ended questions addressed various aspects such as ease of use, understanding the ERD concept, usefulness of feedback, and confidence in ERD design. Some open-ended questions were incorporated, but they supplemented the quantitative analysis and were not part of an independent strand of analysis. Like the

pre-development survey, the instrument was reviewed by faculty experts, pilot-tested with a small group of students, and administered under strict ethical guidelines.

Figure 1 illustrates the overall research methodology at various stages, leading to the development of ERD-PRO, including the pre-development survey phase, development phase, deployment of ERD-PRO, and post-development survey phase, as outlined.

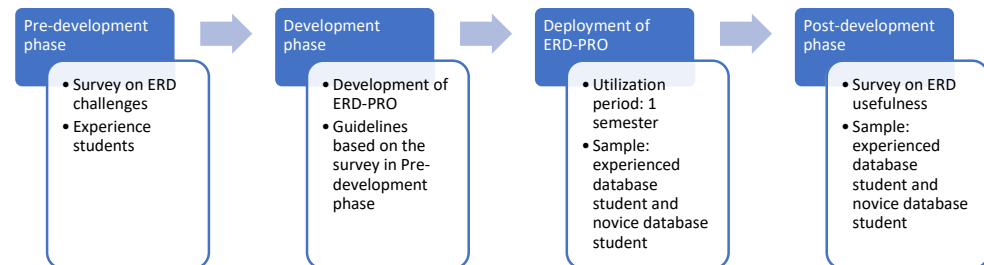


Figure 1. Research Methodology Process Flow.

3.4. Data Analysis

Data analysis followed the mixed-methods design. Interview data prior to development were transcribed and analyzed thematically, with particular attention paid to recurring problems in ERD design such as misinterpretation of composite entities, multi-valued attributes, and many-to-many relationships. Coding revealed patterns of difficulty that directed the ERD-PRO design.

Post-development data were analyzed quantitatively. Survey data were examined with descriptive statistics, including frequencies, percentages, and mean scores, to inform student attitudes toward the effectiveness of ERD-PRO. Novice and advanced learners were contrasted in the hope of establishing experience differences between groups. Statistical tests were selected where appropriate, according to the data distribution, utilizing parametric or non-parametric methods as necessary, to provide a valid interpretation. Although some open-ended comments were obtained, these were kept as example cases and not systematically analyzed, since the main emphasis of the post-development survey was quantitative.

3.5. Survey Instrument

The survey tool was pretested via expert review among faculty members in database education to ascertain content clarity and importance. Piloting of the survey instrument was conducted using a sample of small-sized students prior to the main study to test the items. Ethical practices were strictly observed, involving informed consent, voluntary agreement, and confidentiality of answers.

3.6. Survey Content

The pre-development survey was designed to identify students' challenges in ERD design, covering topics such as common ERD concepts, selection of primary and multivalued attributes, determining suitable relationships, avoiding redundancy, and assessing difficulty levels for various ERD components using Likert scales and multiple-choice questions.

The post-development survey evaluated students' experiences using ERD-PRO, including the usefulness of explanations and reasoning, scenario clarity, challenges encountered, and overall recommendation of the system. Both surveys aimed to capture students' perceptions and experiences, providing insights into the usability and learning impact of ERD-PRO. The full list of survey questions is provided in Appendices A and B.

4. ERD-PRO

This ERD-PRO is expected to be an essential learning tool, transforming the comprehension of the complex realm of Entity Relationship Diagrams (ERDs) with intelligent teaching, explanation, and reasoning algorithms, as well as an easy-to-use design interface. The Intelligent Tutoring System of ERD-PRO aims to be the dynamic educational companion of database students. It is designed to significantly enhance the learning experience by diving into the details of ERD mechanisms. The system serves as a structured and intuitive platform in the learning landscape of ERD, where complexity and ambiguity often hinder understanding. The system has two modes: instructor and student. The student mode is designed to guide users seamlessly through the ERD creation process, offering autonomy to explore and expert assistance as needed. While in instructor mode, the instructors can create sample scenarios of entity relationships to help students understand various aspects through reasoning and explanation.

For instance, the instructor needs to create a code to the scenario, title, difficulty level, and scenario explanation as shown in Figure 2. The title is Learning Management System (LMS), the difficulty level of this scenario is “Difficult”, and the scenario paragraph explains how LMS works. The creation of the scenario is shown in Figure 2. The scenarios were created at the administration level, and these scenarios will be cascaded down to every instructor’s account. Each instructor has the freedom to make changes to the scenario, attributes, entities, data types, business rules, and also the text in the Reason and Explanation. The changes at each instructor level will remain in his/her domain and will not affect the source at the administrator level.

The screenshot shows the 'Assignment' form in the ERD-PRO Instructor Mode. At the top, there is a progress bar with six steps: 1. Scenario, 2. Entities, 3. Attributes, 4. Attribute Types, 5. Relations, and 6. Business Rule. The 'Assignment' form is a modal window with a blue header and a red close button. It contains the following fields: 'Assignment Code *' with the value 'MPROJ330', 'Due Date *' with the value '2024-08-29', 'Title *' with the value 'LMS', and 'Difficulty Level *' with a dropdown menu set to 'Difficult'. Below these is a 'Scenario *' text area containing the text: 'In an online learning management system (LMS), students enroll in various courses, each containing multiple modules and assessments. Instructors manage course content, track student progress, and provide feedback, while administrators oversee user accounts, course availability, and reporting functionalities.' At the bottom of the form is a 'Submit' button.

Figure 2. Scenario Description—Instructor Mode.

In the following step, the instructor needs to create entities. As Figure 3 shows, entities related to LMS have been created, including Student, Courses, Instructors, and Assessments, and each entity is assigned a unique Record ID. The entities, however, can be deleted or edited.

The screenshot shows the 'Assignment Entities List' table in the ERD-PRO Instructor Mode. At the top, there is a progress bar with six steps: 1. Scenario, 2. Entities, 3. Attributes, 4. Attribute Types, 5. Relations, and 6. Business Rule. The 'Assignment Entities List' is a modal window with a blue header and a red close button. It contains a 'Filter' input field and a table with the following data:

Record ID	Record Date	Entity	Actions
1193	28/08/2024	Student	[Edit] [Delete]
1194	28/08/2024	Courses	[Edit] [Delete]
1195	28/08/2024	Instructors	[Edit] [Delete]
1196	28/08/2024	Assessments	[Edit] [Delete]

Figure 3. Describing Entities—Instructor Mode.

Figure 4 illustrates the need to manually provide the reasons along with explanations for each entity. The reason for the Student entity is that it “represents the users who enroll in and participate in courses”, while its explanation is, “individuals who register for courses to gain knowledge and skills in specific subjects”.

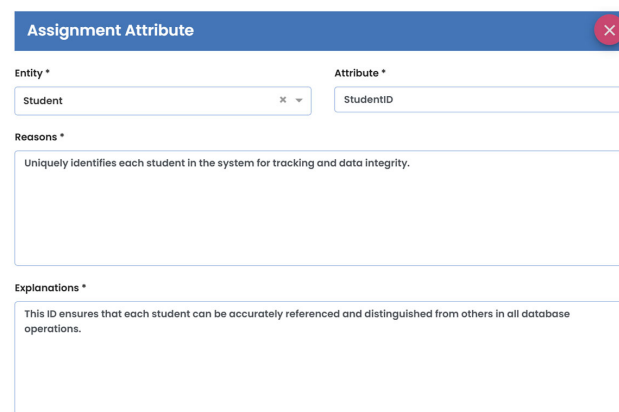
Figure 4. Reason and Explanation about Entities—Instructor Mod.

It then needs to assign attributes as shown in Figure 5. For instance, the attributes for the Student entity include StudentID, Name, Email, and EnrollmentDate.

Record ID	Entity	Attribute	Actions
3611	Student	StudentID	
3612	Student	Name	
3613	Student	Email	
3614	Student	EnrollmentDate	

Figure 5. Attribute Assignment—Instructor Mode.

For each entity, reasons and explanations (Figure 6) are provided for each attribute, allowing students to understand how attributes are determined. For example, StudentID attribute uniquely identifies each student in the system for tracking and data integrity. StudentID ensures that each student can be accurately referenced and distinguished from others in all databases and operations.



Assignment Attribute

Entity *
Student

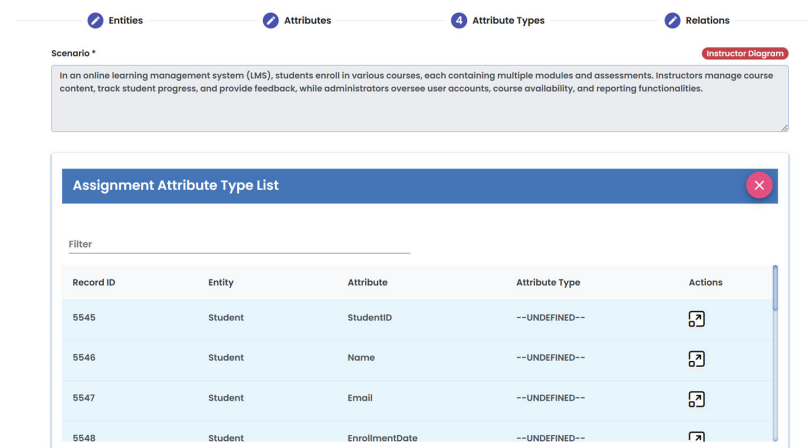
Attribute *
StudentID

Reasons *
Uniquely identifies each student in the system for tracking and data integrity.

Explanations *
This ID ensures that each student can be accurately referenced and distinguished from others in all database operations.

Figure 6. Reasons and Explanations for Attributes—Instructor Mode.

The assignment of a data type to an attribute is optional, as shown in Figure 7. For example, the attribute of StudentID is usually a primary key, often in the form of an integer.



Entities Attributes **Attribute Types** Relations

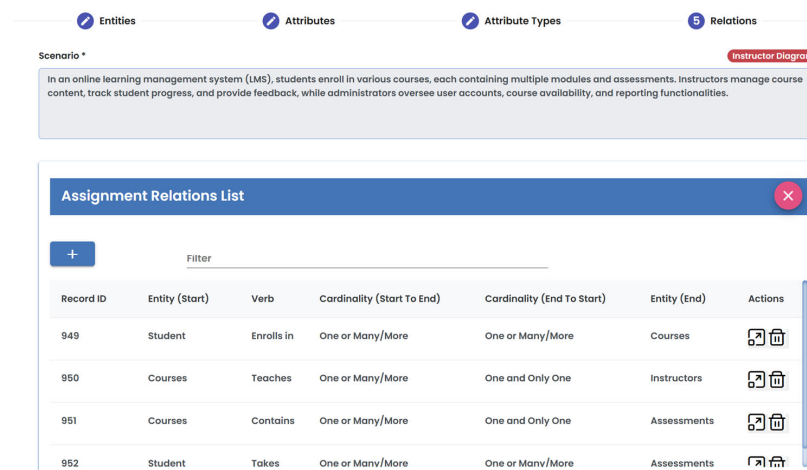
Scenario *
In an online learning management system (LMS), students enroll in various courses, each containing multiple modules and assessments. Instructors manage course content, track student progress, and provide feedback, while administrators oversee user accounts, course availability, and reporting functionalities.

Assignment Attribute Type List

Record ID	Entity	Attribute	Attribute Type	Actions
5545	Student	StudentID	--UNDEFINED--	[Icon]
5546	Student	Name	--UNDEFINED--	[Icon]
5547	Student	Email	--UNDEFINED--	[Icon]
5548	Student	EnrollmentDate	--UNDEFINED--	[Icon]

Figure 7. Attribute Types Assignment—Instructor Mode.

The assignment of attribute type is followed by the assignment of relationships, as shown in Figure 8. For instance, a Student entity can enroll in one or more courses, where a Course is another entity. This means a Student entity can have one or more relationships with a Course entity, and vice versa. While the entity of Course has one and only one relationship with the course, like each Course can be taught by one and only one instructor.



Entities Attributes Attribute Types **Relations**

Scenario *
In an online learning management system (LMS), students enroll in various courses, each containing multiple modules and assessments. Instructors manage course content, track student progress, and provide feedback, while administrators oversee user accounts, course availability, and reporting functionalities.

Assignment Relations List

Record ID	Entity (Start)	Verb	Cardinality (Start To End)	Cardinality (End To Start)	Entity (End)	Actions
949	Student	Enrolls in	One or Many/More	One or Many/More	Courses	[Icon] [Icon]
950	Courses	Teaches	One or Many/More	One and Only One	Instructors	[Icon] [Icon]
951	Courses	Contains	One or Many/More	One and Only One	Assessments	[Icon] [Icon]
952	Student	Takes	One or Many/More	One or Many/More	Assessments	[Icon] [Icon]

Figure 8. Assigning Relations—Instructor Mode.

In ERD-PRO, the instructor must provide an explanation for each relationship, as shown in Figure 9. The reason for “One and Many/More” relationship between Student and Course entities is that “each course is taught by one instructor, but an instructor can teach multiple courses”.

Assignment Relations

Entity (Start) *
Student

Cardinality (Start To End) *
One or Many/More

Cardinality (End To Start) *
One or Many/More

Entity (End) *
Courses

Verb *
Enrolls in

Reasons *
Each course is taught by one instructor, but an instructor can teach multiple courses.

Explanations *
This allows for clear assignment of courses to instructors while facilitating instructor workload management.

Figure 9. Reasons and Explanations while Assigning Relations—Instructor Mode.

Lastly, Figure 10 shows the assignment of business rules, which must be aligned with the ERD itself.

Assignment Business Rule List

Entities Attributes Attribute Types Relations **Business Rule**

Scenario *

In an online learning management system (LMS), students enroll in various courses, each containing multiple modules and assessments. Instructors manage course content, track student progress, and provide feedback, while administrators oversee user accounts, course availability, and reporting functionalities.

Record ID	Business Rule	Actions
-----------	---------------	---------

Figure 10. Assigning Business Rules—Instructor Mode.

In student mode, ERD-PRO focuses on practical application by allowing students to select from various ERD scenarios available (refer to Figure 11), such as library management systems and clinician offices. The system helps students to learn ERD development by bridging the gap between theory and real-world application. Each stage offers hands-on exercise, ensuring concepts are learned, deeply understood, and internalized.

Assignment Entities List

Entities Attributes Attribute Types Relations Business Rule

Scenario *

In an online learning management system (LMS), students enroll in various courses, each containing multiple modules and assessments. Instructors manage course content, track student progress, and provide feedback, while administrators oversee user accounts, course availability, and reporting functionalities.

Record ID	Record Date	Entity	Actions
-----------	-------------	--------	---------

Figure 11. List of Tasks to Perform—Student Mode.

The entity creation stage has a contextual help box that transforms potential road-blocks into learning opportunities. This stage encourages students to generate entities independently (refer to Figure 12) and provide insightful suggestions to foster a dynamic and collaborative learning environment.

Assignment List				
Filter				
Assignment Code	Title	Difficulty Level	Status	Actions
ASGN-02	Cleaning Service Booking	Moderate	Pending	
ASGN-04	Farmer Business	Moderate	Pending	
MPROJS30	LMS	Difficult	Pending	
MPROJS02	Bank loans management system	Basic	InProgress	
MPROJS07	Supermarket Inventory Database	Moderate	InProgress	
MPROJS01	Library Management System	Basic	InProgress	

Figure 12. Assignment of Entities—Student Mode.

An intuitive help box provides entity suggestions with names, reasons, and explanations (refer to Figures 13 and 14). For example, a prompt for the entity “book” includes rational and detailed insights. Students are free to opt for suggested entities or unleash their imagination.

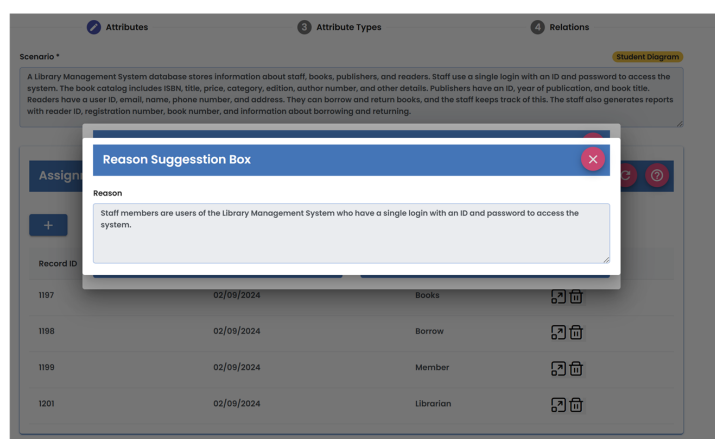


Figure 13. Reason for Suggestion of Certain Entity—Student Mode.

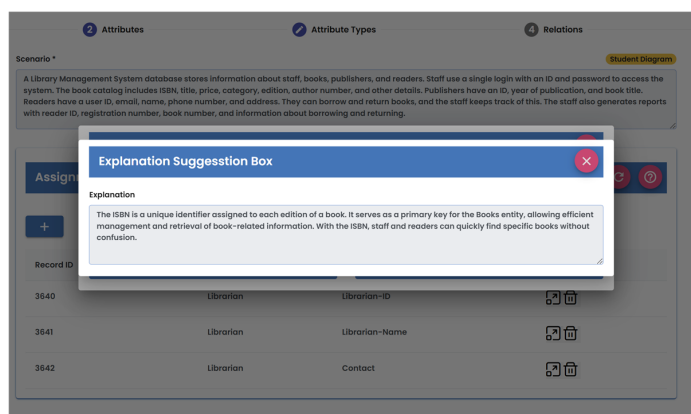


Figure 14. Explanation for Suggestion of Certain Entity—Student Model.

The attribution stage helps students to assign attributes for the chosen identity (refer to Figure 15), or they can request help to get suggestions.

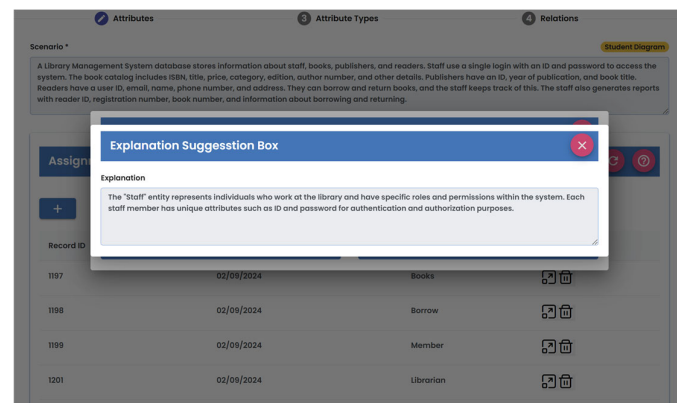


Figure 15. Assignment of Attributes—Student Mode.

The system will offer related suggestions such as ISBN, Author Name, and Book Title alongside readily availability for any uncertainties (Figures 16 and 17).

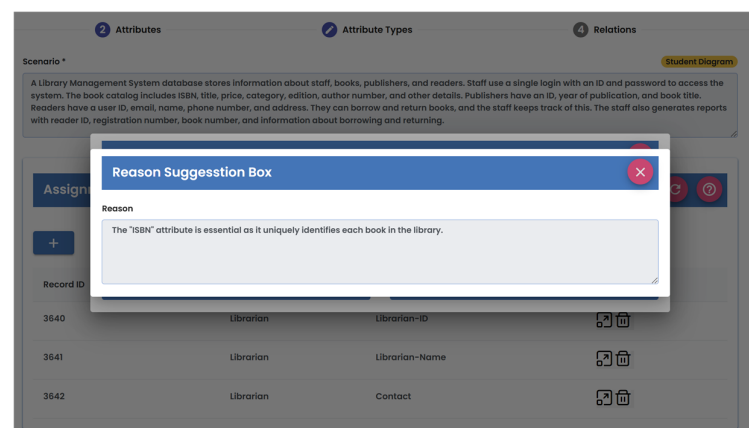


Figure 16. Reason for Suggesting Certain Attribute—Student Mode.

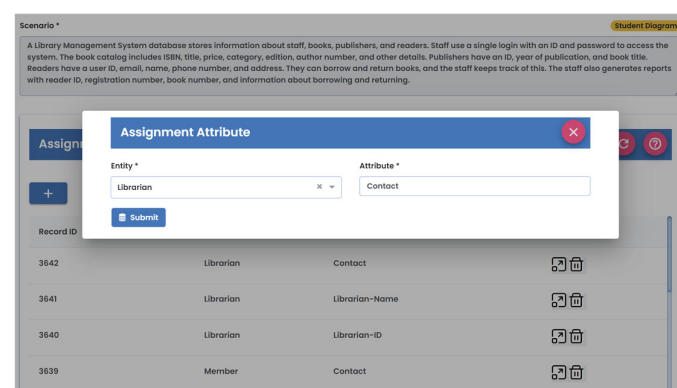


Figure 17. Explanation for Suggesting Certain Attribute—Student Mode.

Moving on to the next stage, students are required to assign specific types to attributes or ask for help to obtain the appropriate types, accompanied by explanations and reasoning to enhance understanding. However, this requirement is optional. Both the attribute creation and type assignment stages offer a tangible aspect to abstract concepts. Subsequently, students need to define the relationship between entities (refer to Figure 18), and this

stage allows them to set cardinalities while offering suggestions. For instance, establish a one-to-many relationship between a book and a publisher by specifying that one book can be associated with only one publisher. Still, a publisher can have multiple related books.

Record ID	Entity (Start)	Relationship	Cardinality (End To Start)	Cardinality (Start To End)	Entity (End)	Actions
953						
954	Member	Borrows	One or Many/More	One and Only One	Borrow	[Icons]
955	Librarian	manages	One or Many/More	One and Only One	Borrow	[Icons]
956	Books	catalogs	One or Many/More	One and Only One	Librarian	[Icons]

Figure 18. Defining Relationships—Student Mode.

Here, students will grasp the theoretical aspects of relationships and can view results through diagrams with and without attributes (as shown in Figure 19). Lastly, ERD-PRO requires students to set business rules, which is also an optional task.

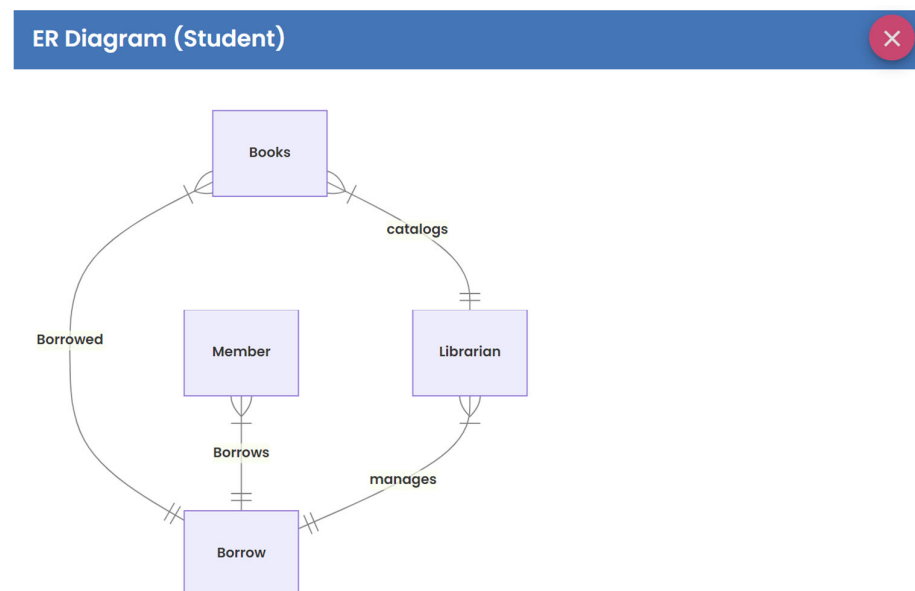


Figure 19. Automatically Created ERD based on defined entities and their relationships.

In short, ERD-PRO not only focuses on coaching but also empowers students to apply knowledge by creating entities and attributes, assigning types, and mirroring their decisions in a real-world scenario. Therefore, it prompts critical thinking and decision-making by applying the acquired knowledge in a practical context. In short, it is a transformative tool that equips students with the theoretical knowledge and practical expertise needed to navigate the complexities of ERD creation with confidence and proficiency.

ERD-PRO Framework with XAI Concept

As the framework in Figure 20 describes, the tutor creates scenarios and appropriate design elements for each scenario, including entities and cardinalities. The student can use these scenarios or create new ones to practice ERD. The student's interaction will be saved in the database, and these files must be transferred to an external XAI system for analysis. The tutor will analyze the students' progress reports and formulate policies to improve.

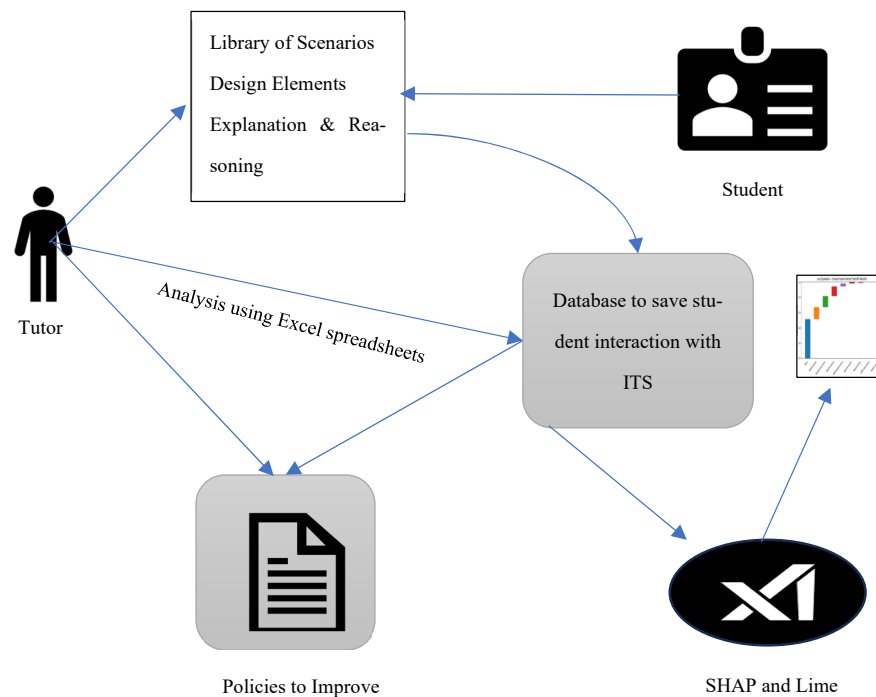


Figure 20. ERD PRO Framework.

5. Research Findings and Discussion

5.1. Pre-Development Survey

The pre-development survey employed open-ended questions in order to gather students' experiences and challenges with ERD design. Comments were theme-coded and grouped through thematic analysis, where common issues were tagged and grouped. One student responded, for example, "I'm confused between composite attributes and entities—I'm never sure which one to apply". Another student stated, "I don't fully see why we need bridge entities; I usually just join the tables together".

Through the responses, there were similarities: (1) difficulties in understanding composite objects, bridge objects, and multivalued attributes; (2) difficulties in properly specifying many-to-many relations; and (3) misunderstandings about primary key constraints and the elimination of redundancy. These findings suggest that more dynamic and interactive teaching techniques are the solution, as conventional teaching often fosters one-way communication and limits discovery.

5.2. Post-Development Findings

According to the findings of a preliminary assessment, common problems included confusion in identifying entities and relationships, avoiding redundancy in relationships, and difficulty working with complex concepts such as composite entities and multivalued attributes. It therefore guided the design and functionality of ERD-PRO by incorporating targeted features such as step-by-step guidance, immediate feedback, and practice scenarios specifically addressing the challenges identified in the pre-development survey. Analysis of

the feedback provided by students after the implementation of ERD-PRO suggests potential improvements in their learning process and confidence in ERD design. It is important to note that the data collected reflects students' perceptions of the tool's usefulness rather than objective measures of learning outcomes. The same cohort of students participated in both the pre-development survey and the post-implementation feedback, allowing us to track their reported experiences and perceived growth. While these subjective assessments indicate that ERD-PRO was generally well-received, with 50% of students rating it as "Somewhat easy to use" in helping them understand the main ideas of ERDs. While 33% called it "Easy to Use" and 17% said it's "Not Easy to Use" (Figure 21). Therefore, further studies using objective performance metrics would be needed to confirm its actual effectiveness in improving ERD skills.

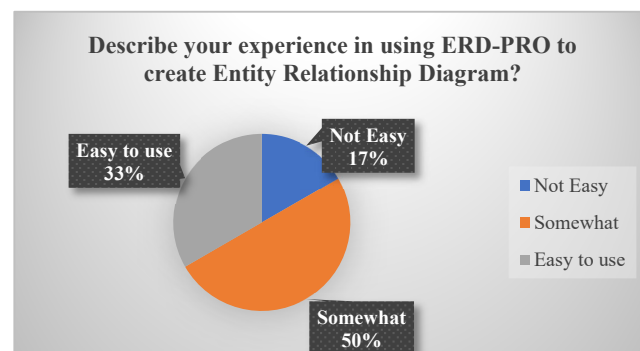


Figure 21. Overall experience of students with ERD-PRO.

Figure 22 shows students' self-reported understanding of complex ERD concepts after using ERD-PRO. Responses on the Likert scale were: 7% 'Not Improved at All', 13% 'Not Improved', 33% 'Neutral', and 47% 'Improved' or 'Significantly Improved'. These results highlight students' perceptions of the tool's usefulness in supporting their analytical approach, although further evaluation with objective measures would be necessary to confirm actual learning gains.

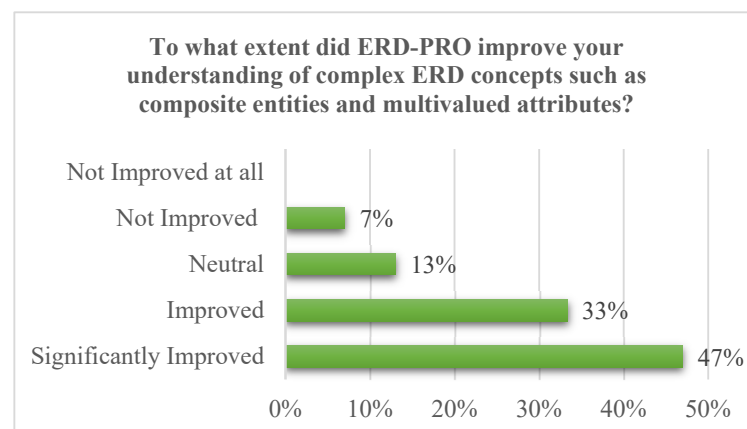


Figure 22. Improvement in Understanding ERD Concepts.

The third question aimed to determine the extent to which the explanations and reasoning provided by ERD-PRO aided students in understanding complex ERD concepts, such as composite entities and multivalued attributes. A strong majority (63%) of the students reported that the explanations and reasoning features either "Significantly Improved" or "Improved" their understanding of these concepts (Figure 23). These findings suggest that students perceived the explanatory features of ERD-PRO as helpful in making complex

ERD topics more approachable. However, further empirical studies would be required to determine whether these perceptions translate into measurable improvements in deep learning of advanced ERD components.

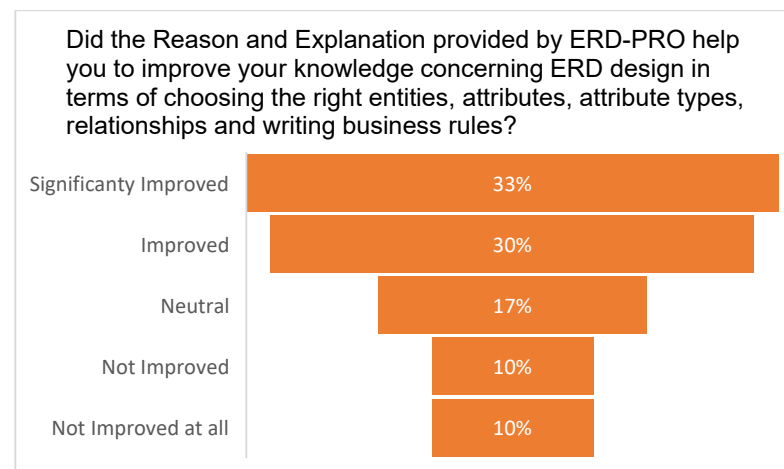


Figure 23. Effectiveness of Reasoning and Explanation.

In one of the questions, students were asked how confident they felt in designing ERDs after using ERD-PRO compared to their prior experience. Results showed that 80% of the students reported feeling “Much Confident” or “Confident” in their ability to design ERDs after using the software (Figure 24). These findings indicate that ERD-PRO was perceived by the majority of students as contributing to an increased sense of confidence in ERD design. However, it is important to note that this result reflects students’ self-reported confidence and does not directly measure their ability to independently apply ERD concepts.

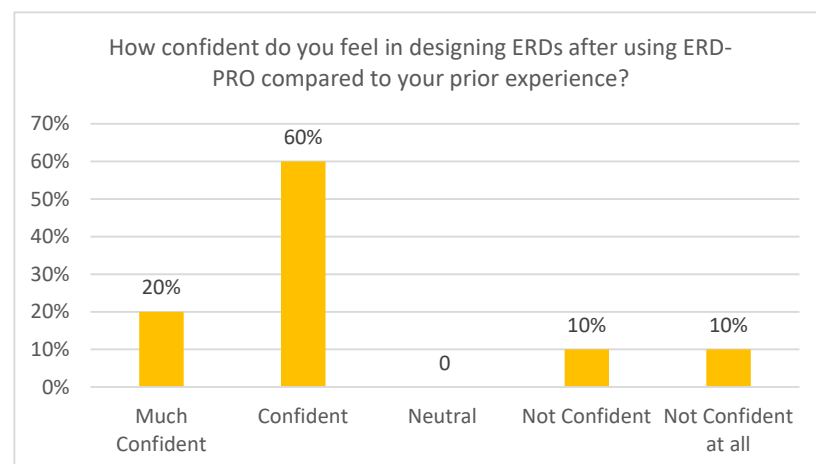


Figure 24. Improvement in Confidence Level.

When students were asked, “How well did ERD-PRO address the challenges you previously faced in ERD design such as understanding of relationships and redundancy avoidance?”, about 65% reported that ERD-PRO “Completely Addressed” or “Mostly Addressed” these specific difficulties (Figure 25). This positive feedback suggests that ERD-PRO may be useful in helping students overcome challenges particularly related to understanding relationships and avoiding redundancy in ERD design. However, these findings cannot be generalized to all possible ERD challenges, as the question specifically focused on these two areas.

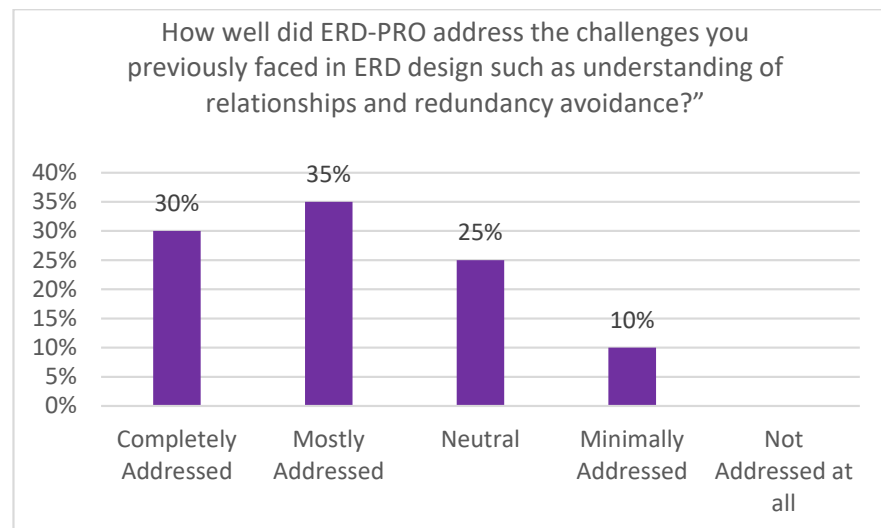


Figure 25. Addressing challenges related to ERD understanding.

Students were asked if they thought the step-by-step guidance given to them by ERD-PRO was helpful for them in the process of learning. Approximately 75% of the students believe that “Strongly Agreed” or “Agreed” step-by-step guidance was helpful. It can be observed that students also appreciate the guided learning approach followed by ERD-PRO. It thus suggests that with the structured and incremental approach toward learning, their comprehension is better in keeping the complexities of ERD design (Figure 26).

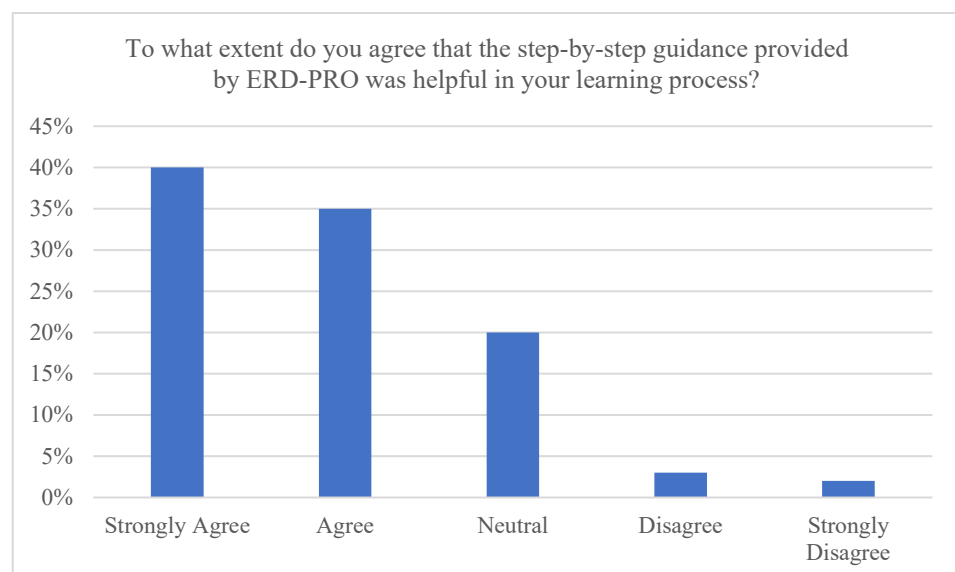


Figure 26. Step-by-step guidance provided by ERD-PRO.

Participating students were asked to respond to the degree ERD-PRO helped them apply ERD concepts to real database design problems. About 65% of the students responded that this helped them either “Greatly Helped” or “Helped” as shown in Figure 27. This result indicates whether ERD-PRO is effective in bridging the gap between theoretical knowledge and practical applications. The positive impact on solving problems from real life suggests that this tool contains relevant and practical knowledge that students can transfer to real database design tasks.

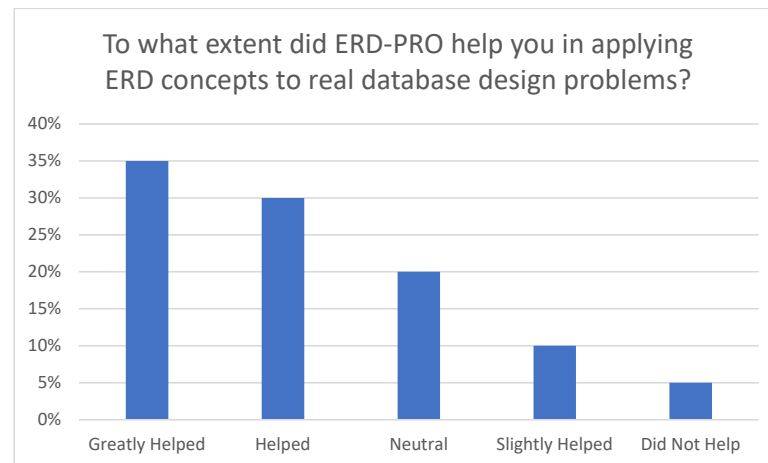


Figure 27. Extent of ERD-PRO in applying ERD concepts to real database design problems.

They were also asked to describe the feedback provided by ERD-PRO, which was used to assess the effectiveness of the given hints, suggestions, and corrections in improving their understanding and skills in ERD design. About 70% described the feedback as “Extremely Useful” or “Very Useful” as shown in Figure 28. Positive feedback indicates that the corrective and guiding abilities of ERD-PRO effectively reinforce learning. Arguably, the most important aspect for students is that feedback is immediate, helping them correct mistakes by providing an explanation of the correct solution. Moreover, one student said, “the suggestions help him clear his doubts and ambiguities about assigning attributes to entities”. It was easier to establish relationships between entities because these suggestions facilitated the creation of an ideal system diagram without external interference from the student. Another student described the experience of learning new things about ERD and its various aspects, including the requirements for assigning attributes and their types. They believed that repeatedly using the tool’s help added to their understanding and efficiency in creating complex ERDs because it reinforces learning and concept development.

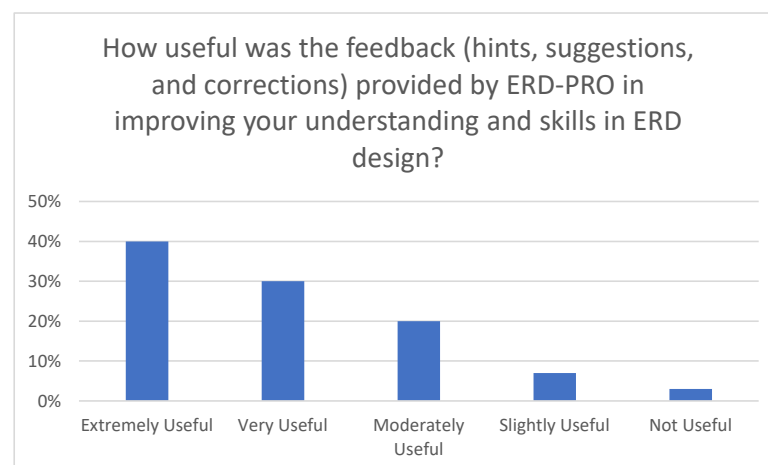


Figure 28. Usefulness of feedback in improving your understanding and skills in ERD design?.

Participating students queried how likely they were going to recommend ERD-PRO to other students for learning ERD concepts. Over 90% of students indicated they were “Very Likely” or “Likely” to recommend ERD-PRO to others (Figure 29). The strong likelihood of recommendation reflects high overall satisfaction with ERD-PRO. This suggests that students believe the software is a valuable resource for others seeking to learn ERD concepts, indicating a strong endorsement of its educational benefits.

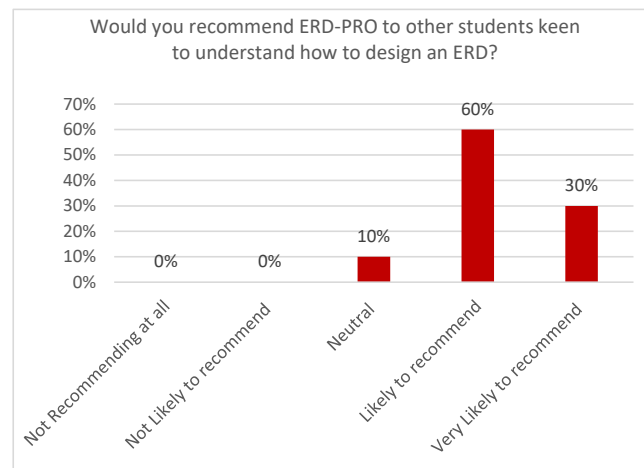


Figure 29. Likely recommending ERD-PRO to others.

The last question of the survey aimed to investigate how frequently students would use ERD-PRO if it were incorporated into database courses. Results showed that 50% of respondents indicated they would use it “Frequently”, 17% “Very Frequently”, and 33% “Occasionally”, with no respondents selecting “Rarely” or “Never” (Figure 30). These responses highlight students’ perceptions of ERD-PRO’s potential usefulness in supporting database learning. However, it is important to note that these are self-reported intentions of future use rather than data on actual usage patterns. Additional system log data would be needed to substantiate the frequency of actual use and to compare it with students’ perceived frequency.

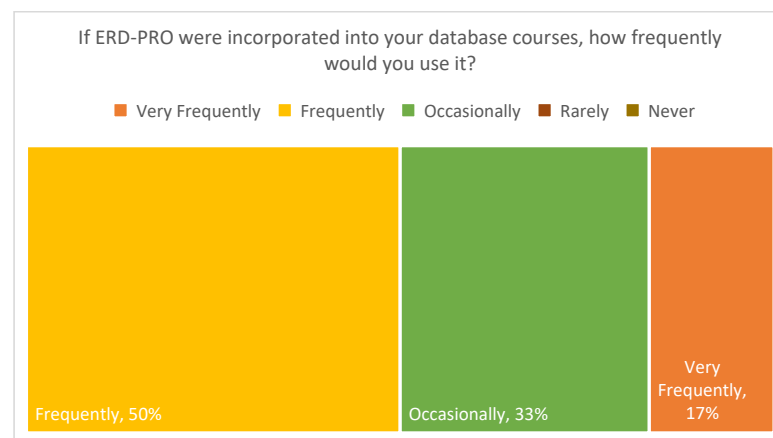


Figure 30. Usage Frequency.

6. Implications for Practitioners

The findings of this study have several important implications for instructors teaching database concepts. First, ERD-PRO’s incremental progress and reasoning capabilities can be incorporated into database classes as a supplementary tool, allowing instructors to transition from one-way lecturing to more exploratory and interactive learning environments. The positive student feedback indicates that the integration of intelligent tutoring systems can significantly enhance understanding of complex ERD concepts, particularly in subject matter areas where misconceptions are likely to prevail (e.g., composite objects, relationships, and redundancy avoidance). Second, the teacher mode in ERD-PRO allows educators to define individualized scenarios according to their own curriculum, providing tailored practice experience targeted at individual student weaknesses. Finally, through the encouragement of immediate feedback and independent discovery, ERD-PRO supports ac-

tive learning methods of database design that can promote heightened student confidence, retention of skills, and eventual proficiency.

6.1. Student Activities and XAI Interpretation

The ERD-PRO system records every student's interaction while students attempt to design Entity-Relationship Diagrams (ERDs). Each movement or click, such as selecting entities, defining attributes, clicking the Reason or Explanation buttons, drawing relationships, or applying business rules, is automatically logged and stored in an Excel file. For example, attributes like TotalScenarios, TotalDuration, BasicScenario, ModerateScenario, DifficultScenario, ReasonClicks, and ExplanationClicks are captured for each attempt. At the end of a session, the instructor downloads an Excel file containing a detailed report of student activities. These files are then compiled manually, and students who manage to complete the targeted number of scenarios, reaching the stage of generating a complete ERD diagram, are marked as Complete. All student data is eventually consolidated into a single dataset for further analysis using Explainable AI (XAI) techniques such as SHAP and LIME (See Appendix C for full coding).

6.2. XAI Interpretation

Global Feature Importance (SHAP): The global SHAP in Figure 31 shows that ExplanationClicks and ReasonClicks are the most influential attributes determining whether a student successfully completes their ERD task. This suggests that students who actively seek explanations and reasoning aids are more likely to succeed. Difficulty level, particularly DifficultScenario, also contributes significantly, reflecting that advanced tasks strongly influence completion outcomes. Other features, such as TotalDuration, ModerateScenario, and TotalScenarios, have smaller but still notable impacts, while BasicScenario plays a minimal role.

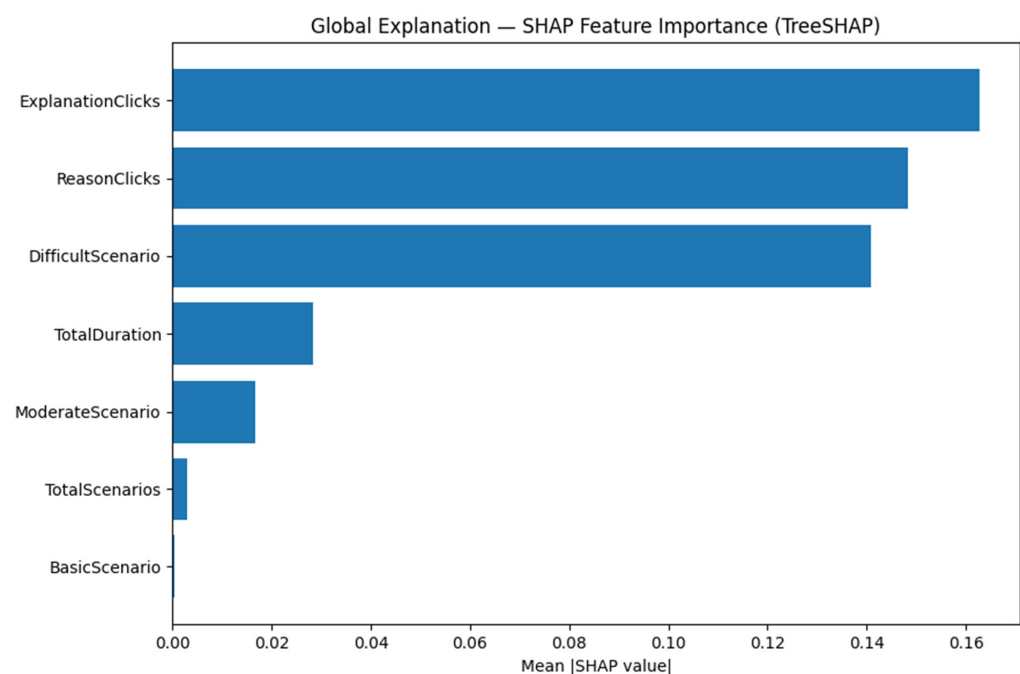


Figure 31. Global SHAP.

Local Waterfall Explanation (TreeSHAP): Figures 32–34 display the local SHAP waterfall plot for an individual student's attempt. Starting from the model's baseline expectation, positive contributions from ExplanationClicks, ReasonClicks, and DifficultScenario push the prediction towards completion, while minor contributions from TotalDuration and

ModerateScenario provide additional support. This breakdown illustrates how various features collectively impact one student’s specific result.

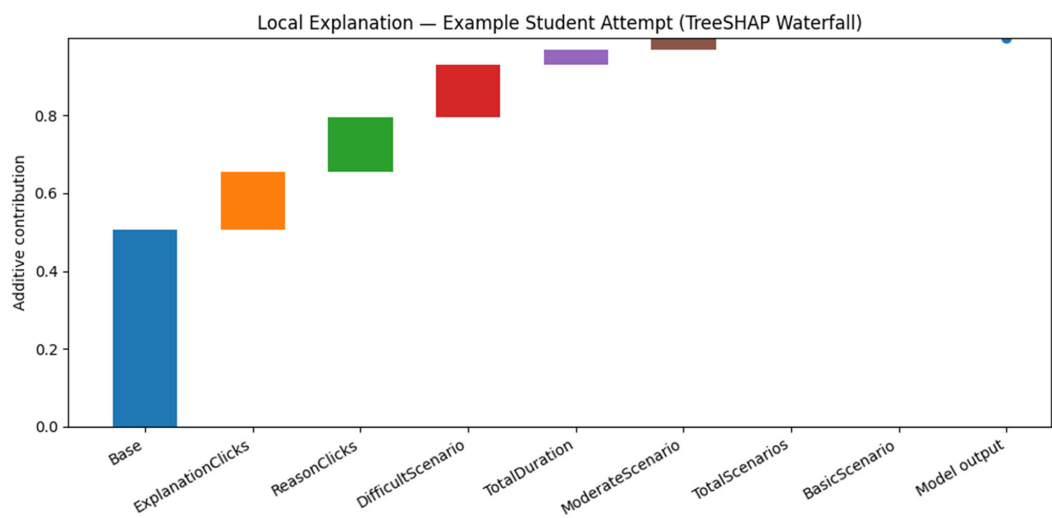


Figure 32. SHAP for Student A.

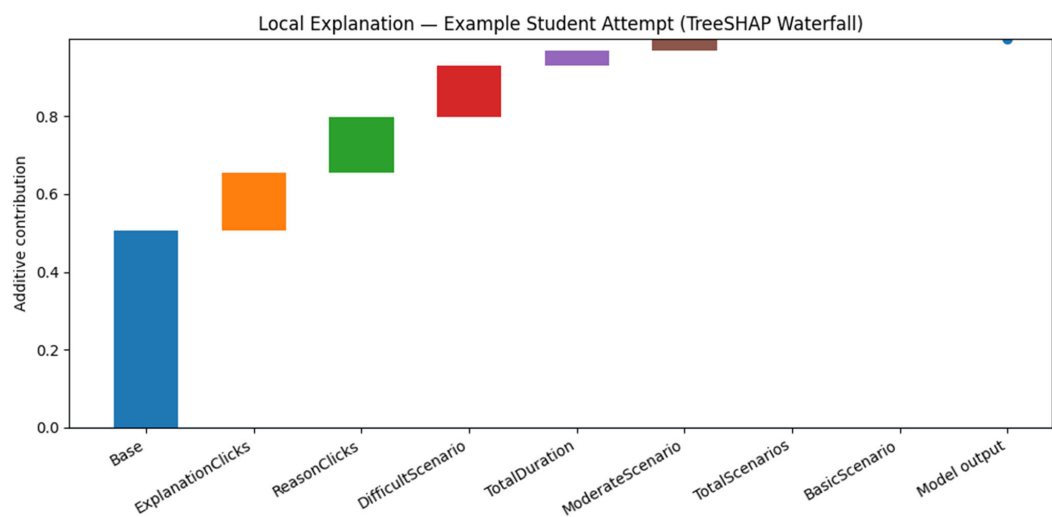


Figure 33. Local SHAP for Student B.

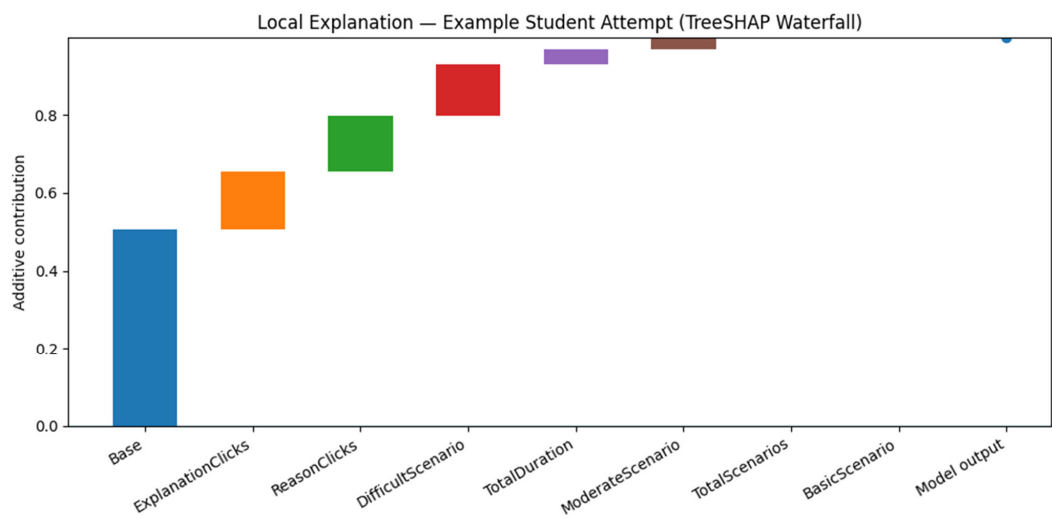


Figure 34. SHAP for Student C.

Local Weights (LIME): The LIME tables provide an alternative perspective on local interpretability as shown in Figures 35–37. For example, a student with ExplanationClicks greater than 19.75 and ReasonClicks greater than 22.75 receives positive weights, indicating that these behaviors strongly increase the likelihood of completion. Similarly, handling DifficultScenario tasks within certain thresholds contributes positively, while excessively high numbers of scenarios or minimal engagement may negatively affect the prediction. These insights help instructors understand which exact actions influenced each student’s outcome.

Local Explanation — Example Student Attempt (LIME Weights)

Local feature	Weight
ExplanationClicks > 19.75	0.23155663084069017
ReasonClicks > 22.75	0.2125018210999828
0.00 < DifficultScenario <= 2.75	0.16763241026109194
TotalDuration > 202.50	0.05192428455907531
ModerateScenario > 2.00	0.025046757124658047
7.50 < TotalScenarios <= 9.00	−0.008142647349858785

Figure 35. LIME for Student A.

Local Explanation — Example Student Attempt (LIME Weights)

Local feature	Weight
ExplanationClicks > 19.75	0.23812685379065965
ReasonClicks > 22.75	0.2170483588527876
0.00 < DifficultScenario <= 2.75	0.18710647210459383
TotalDuration > 202.50	0.04918863787962917
ModerateScenario > 2.00	0.029963229474074534
BasicScenario <= 3.00	−0.008236876195987375

Figure 36. LIME for Student B.

Local Explanation — Example Student Attempt (LIME Weights)

Local feature	Weight
3.00 < ExplanationClicks <= 19.75	0.2200495174497004
ReasonClicks > 22.75	0.21705444301165142
0.00 < DifficultScenario <= 2.75	0.17471282516454667
TotalDuration > 202.50	0.04295089831487642
ModerateScenario > 2.00	0.028962756988982312
BasicScenario <= 3.00	0.010182802874299252

Figure 37. LIME for Student C.

7. Discussion

The results of this study indicate that the goals of ERD-PRO development and evaluation as an Intelligent Tutoring System (ITS) were successfully achieved. The primary goal, studying the most critical issues students face when creating Entity-Relationship Diagrams (ERDs), was addressed in the pre-development survey. Students identified issues related to identifying entities and relations, handling multivalued or composite attributes, and

applying business rules consistently. These findings are in line with earlier work identifying normal learning impediments in database classes, where students find it difficult to connect ERD abstract notions to concrete models [24,35]. It was through the identification of these flaws that the study confirmed the need for an adaptive ITS with contextualized, guided support.

ERD-PRO's design squarely addressed these problems. Post-development survey results reflected meaningful improvement in students' understanding and analytical skills: 80% of the students reported enhanced ability to identify and distinguish between entities, attributes, and relationships. Additionally, 65% of the students reported that the explanation and reasoning facilities in the system had improved their understanding of ERD concepts. These findings are consistent with studies emphasizing the effectiveness of ITS with Explainable AI (XAI) in enabling active learning, self-explanation, and reflective thinking [20,21,25]. Through providing step-by-step directions, immediate feedback, and justification of decisions, ERD-PRO acted as a "more knowledgeable other" within the Zone of Proximal Development framework, maintaining Vygotsky's constructivist agenda [42].

These post-development results also indicate a significant increase in students' confidence levels, with 80% reporting being confident or much more confident in creating ERDs. High satisfaction levels, with 90% of students volunteering to recommend the system, validate that ERD-PRO not only facilitated easy learning but also engaged and motivated the learners. This aligns with previous ITS studies emphasizing personalized feedback and interactive guidance as being crucial in augmenting learning outcomes in STEM learning [19,22,23].

In conjunction with ERD learning, these results suggest that the ERD-PRO framework can be applied to guide ITS development in other high-complexity areas, such as mathematics, computer programming, and language learning. Through the adoption of mixed-initiative strategies and explanation and reasoning modules on the basis of XAI, such systems can assist students to solve intricate problem-solving tasks, promote self-regulated learning, and bridge the theoretical-practical gap [21,30,31].

In summary, ERD-PRO demonstrates that incorporating well-designed ITS and explainable reasoning, along with guided structuring, can effectively remediate common learning problems, enhance conceptual understanding, and boost learner confidence. This research contributes to the overall literature by offering empirical results that reveal that XAI-enhanced ITS is not only able to extend knowledge acquisition but also extend analytical skill and learner self-efficacy, closing gaps that were identified in previous ITS studies [24,25,35]. Subsequent research can attempt to scale this procedure for application with larger and more diverse cohorts of students and other disciplines that require sophisticated conceptual thinking.

8. Conclusions

This study aimed to resolve the problems that have confronted students in ERD design and assess the efficacy of ERD-PRO, an XAI-based mixed-initiative intelligent tutoring system. The experiment verifies the following points: ERD-PRO significantly improved students' ability to identify entities, attributes, and relationships, as well as their analytical capability and confidence level in ERD design. Misconceptions had been resolved, and a deeper understanding gained through explanation and reasoning capabilities.

Despite these positive outcomes, the study has a few limitations. The evaluation was limited to a single student group and a single academic setting, which may restrict the generalizability of the outcomes. Additionally, the concern was largely with short-term performance enhancement, with longer-term retention and applicability not being examined.

Future research should extend this work by piloting ERD-PRO in diverse institutions and subjects, exploring longer-term learning impacts, and pursuing the extension of XAI-based tutoring to additional knowledge domains such as programming, mathematics, and language learning. Following these lines of inquiry will provide valuable insight into the scalability and flexibility of intelligent tutoring systems.

Funding: Zayed University under RIF R22046.

Data Availability Statement: Data is available upon request.

Conflicts of Interest: The author declares no conflict of interest.

Appendix A. Pre-Development Survey



Pre Development – Survey

Q1: The following are common concepts to develop ERD.

Q2: what is the suitable attribute to become a primary key constraint for the table staff?

Q3: what is the suitable attribute to be multivalued attribute in the staff table?

Q4: for each case there are lots of lawyers who will defend the case, each lawyer defend many cases at the same time. What is the best relationship for this statement?

Q5: can you include redundant representations when designing an ERD? For example, the name “StaffName” twice: once as a related entity and once as an attribute.

Q6: How difficult do you find it to identify the following ERD components?

- Entity
- Composite entity / Bridge entity
- Attribute
- Data type

(Likert scale: Very Easy – Easy – Neutral – Difficult – Very Difficult)

Q7: What difficulties do you face in identifying the relationships between entities in ERDs?

- Misunderstanding the meaning of certain relationships
- Don’t know how to link entities with the correct relationship
- Cannot differentiate between weak and strong relationships
- Don’t know a suitable relationship type for entities
- No difficulty in identifying relationships *(Multiple choice, select all that apply)*

Q8: Do you have any other issue or problem in learning and constructing the ER Diagram?

Appendix B. Post-Development Survey

Post Development – Survey

Welcome to our survey on evaluating the ERD-PRO system. It is believed that you have completed a minimum of 8 scenarios in the ERD-PRO prior to taking this survey (indicate your choice below). By participating in this survey, you consent to share your opinions and experiences with us. Please note that your participation is entirely voluntary, and you may withdraw from the survey at any point without penalty. Your responses will remain confidential and will only be used for research purposes. Thank you for taking the time to participate in this survey.

I acknowledge that I have completed:

- ☒ 8 or more scenarios
☐ less than 8 scenarios

Optional: I am willing to provide more information or elaborate further on my responses through my _____ (email) or contact number _____

Following could be the possible survey questions.

(CIT 365 and CIT 470 students)

- Q1: Describe your experience in using ERD-PRO to create Entity Relationship Diagram?
- Q2: To what extent did ERD-PRO improve your understanding of complex ERD concepts such as composite entities and multivalued attributes?
- Q3: Did the Reason and Explanation provided by ERD-PRO help you to improve your knowledge concerning ERD design in terms of choosing the right entities, attributes, attribute types, relationships and writing business rules?
- Q4: How confident do you feel in designing ERDs after using ERD-PRO compared to your prior experience?
- Q5: How well did ERD-PRO address the challenges you previously faced in ERD design such as understanding of relationships and redundancy avoidance?"
- Q6: To what extent do you agree that the step-by-step guidance provided by ERD-PRO was helpful in your learning process?
- Q7: To what extent did ERD-PRO help you in applying ERD concepts to real database design problems?
- Q8: How useful was the feedback (hints, suggestions, and corrections) provided by ERD-PRO in improving your understanding and skills in ERD design?
- Q9: Would you recommend ERD-PRO to other students keen to understand how to design an ERD?
- Q10: If ERD-PRO were incorporated into your database courses, how frequently would you use it?

Appendix C. XAI Source Code for SHAP and LIME

```
# -*- coding: utf-8 -*-
"""Another copy of ERD_PRO_XAI_Colab.ipynb

Automatically generated by Colab.

Original file is located at
https://colab.research.google.com/drive/1e0xGTiT7wVtjd-u9mEsUI3gC4KElyEr9

# ERD-PRO XAI (Colab): TreeSHAP + LIME
Train a **Random Forest** on ERD-PRO telemetry, compute **TreeSHAP** global/local explan

**CSV schema (example)**:
`StudentID, ScenarioNo, Difficulty, NumEntities, NumAttributes, NumRelationships, NumBus
"""

!pip -q install shap lime scikit-learn matplotlib pandas numpy

from google.colab import files
print(" Upload your ERD-PRO CSV (e.g., erdpro_data.csv)")
up = files.upload()
CSV_PATH = list(up.keys())[0] # use the uploaded file
print("Using:", CSV_PATH)

import pandas as pd, numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
import os

# --- Config ---
LABEL_COL = "Complete" # target column (1/0)
DROP_ID_COLS = ["StudentId", "StudentName", "Email"] # non-numeric identifiers to drop
OUT_DIR = "xai_outputs"
os.makedirs(OUT_DIR, exist_ok=True)

# --- Load and prepare ---
df = pd.read_csv(CSV_PATH)
assert LABEL_COL in df.columns, f"{LABEL_COL} not found; columns={list(df.columns)}"

X = df.drop(columns=[c for c in [LABEL_COL]+DROP_ID_COLS if c in df.columns])
X = X.select_dtypes(include=['number']).copy() # keep numeric features only
y = df[LABEL_COL]

print("X columns used:", list(X.columns))
print("Label counts:", y.value_counts(dropna=False).to_dict())

# Train/test split
strat = y if y.nunique() > 1 else None
X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.35, random_state=42, stratify=strat
)

# Model
rf = RandomForestClassifier(
    n_estimators=500, max_depth=10, class_weight='balanced', random_state=42
)
rf.fit(X_train, y_train)

# Report
rep = classification_report(y_test, rf.predict(X_test), output_dict=True, zero_division=
pd.DataFrame(rep).to_csv(f"{OUT_DIR}/classification_report.csv")
```

```

print(" Model trained. Report saved to", f"{OUT_DIR}/classification_report.csv")

import shap, numpy as np, matplotlib.pyplot as plt

explainer = shap.TreeExplainer(rf)
shap_values = explainer.shap_values(X_test) # MUST use the same X_test

def mean_abs_shap_per_feature(shap_values, X_test):
    """Return (n_features,) mean |SHAP| robustly across 2D/3D/list shapes."""
    if isinstance(shap_values, list):
        mats = [np.abs(v).mean(axis=0) for v in shap_values] # each (n_features,)
        return np.sum(np.vstack(mats), axis=0)
    sv = np.asarray(shap_values)
    if sv.ndim == 2: # (n_samples, n_features)
        return np.abs(sv).mean(axis=0)
    if sv.ndim == 3: # (n_samples, n_classes, n_features) OR (n_samples
        n_features = X_test.shape[1]
        a, b = sv.shape[1], sv.shape[2]
        if a == n_features: # (n_samples, n_features, n_classes)
            return np.abs(sv).mean(axis=0).mean(axis=1)
        elif b == n_features: # (n_samples, n_classes, n_features)
            return np.abs(sv).mean(axis=0).mean(axis=0)
        else:
            raise ValueError(f"Cannot identify feature axis in SHAP array shape {sv.shape}")
    raise ValueError(f"Unexpected shap_values shape/type: {type(shap_values)} {getattr(n

abs_mean = mean_abs_shap_per_feature(shap_values, X_test) # (n_features,)
feat_names = np.asarray(X_test.columns)
assert abs_mean.shape[0] == feat_names.shape[0]

order = np.argsort(abs_mean)
feat_sorted = feat_names[order]
vals_sorted = abs_mean[order]

plt.figure(figsize=(9,6))
plt.barh(feat_sorted, vals_sorted)
plt.xlabel('Mean |SHAP value|')
plt.title('Global Explanation – SHAP Feature Importance (TreeSHAP)')
plt.tight_layout()
plt.savefig(f"{OUT_DIR}/shap_global_bar.png", dpi=300, bbox_inches="tight")
plt.savefig(f"{OUT_DIR}/shap_global_bar.pdf", dpi=300, bbox_inches="tight")
plt.show()

pd.DataFrame({"Feature": feat_sorted, "MeanAbsSHAP": vals_sorted}).to_csv(
    f"{OUT_DIR}/shap_global_values.csv", index=False
)
print("SHAP global saved to", OUT_DIR)

# 5) SHAP local waterfalls (top-3 most "Complete" predictions) – SHAPE ROBUST
import numpy as np, matplotlib.pyplot as plt

# choose top-3 by P(Complete=1)
class_index_1 = list(rf.classes_).index(1) if 1 in rf.classes_ else 0
prob_complete = rf.predict_proba(X_test)[:, class_index_1]
order_idx = np.argsort(-prob_complete) # descending

def _contrib_1d_for_class(explainer, X_row, class_idx):
    """
    Return (contrib_1d, base_value) with contrib_1d.shape == (n_features,)
    Works for list / 2D / 3D shap_values across SHAP versions.
    """

```

```

sv = explainer.shap_values(X_row)
base = explainer.expected_value

if isinstance(sv, list):
    # One array per class; take the requested class
    sv_cls = np.asarray(sv[class_idx])          # (1, n_features) or (n_features,)
    contrib = np.squeeze(sv_cls)                # -> (n_features,)
    if contrib.ndim != 1:
        # sometimes (1, n_features); squeeze again defensively
        contrib = np.squeeze(contrib)
    base_val = base[class_idx] if isinstance(base, (list, np.ndarray)) else base
    return contrib, float(np.asarray(base_val).squeeze())

arr = np.asarray(sv)
# Possible shapes: (1, n_features), (1, n_classes, n_features), (1, n_features, n_cl
if arr.ndim == 2:
    # (1, n_features)
    contrib = arr[0]
elif arr.ndim == 3:
    _, a, b = arr.shape
    n_features = X_row.shape[1]
    if a == n_features:          # (1, n_features, n_classes)
        contrib = arr[0, :, class_idx]
    elif b == n_features:       # (1, n_classes, n_features)
        contrib = arr[0, class_idx, :]
    else:
        raise ValueError(f"Unexpected SHAP shape {arr.shape} vs n_features={n_featur
else:
    contrib = np.squeeze(arr)

if contrib.ndim != 1:
    contrib = np.squeeze(contrib)
base_val = base if np.isscalar(base) else np.asarray(base).ravel()[min(class_idx, np
return contrib, float(np.asarray(base_val).squeeze())

def shap_waterfall_plot(explainer, X_row, k, class_idx):
    contrib, base_val = _contrib_id_for_class(explainer, X_row, class_idx) # -> (n_feat
    # Pair (feature_name, contribution) and take top 10 by |contribution|
    pairs = sorted(zip(X_row.columns, contrib), key=lambda t: abs(float(t[1])), reverse=
    labels = ['Base'] + [p[0] for p in pairs] + ['Model output']

    plt.figure(figsize=(10,5))
    plt.axhline(0, linewidth=1)
    start = base_val
    plt.bar(0, base_val, width=0.6)
    for i, (_, v) in enumerate(pairs, start=1):
        v = float(v)
        plt.bar(i, v, bottom=start, width=0.6)
        start += v
    plt.plot([len(labels)-1], [start], marker='o')
    plt.xticks(range(len(labels)), labels, rotation=30, ha='right')
    plt.ylabel('Additive contribution')
    plt.title('Local Explanation – Example Student Attempt (TreeSHAP Waterfall)')
    plt.tight_layout()
    plt.savefig(f"{OUT_DIR}/shap_local_waterfall_{k+1}.png", dpi=300, bbox_inches="tight")
    plt.savefig(f"{OUT_DIR}/shap_local_waterfall_{k+1}.pdf", dpi=300, bbox_inches="tight")
    plt.show()

for k, idx in enumerate(order_idx[:3]):
    shap_waterfall_plot(explainer, X_test.iloc[[idx]], k, class_index_1)

```

```

print(" SHAP local waterfalls saved.")

!pip install lime

from lime.lime_tabular import LimeTabularExplainer
import pandas as pd, numpy as np, matplotlib.pyplot as plt

class_names = [str(c) for c in rf.classes_] # use model's class order

expl = LimeTabularExplainer(
    training_data=X_train.to_numpy(),
    feature_names=X_train.columns.tolist(),
    class_names=class_names,
    mode="classification",
    discretize_continuous=True, # set False if you prefer raw features
    kernel_width=3.0,
    random_state=42
)

for k, idx in enumerate(order_idx[:3]):
    x_row = X_test.iloc[idx].to_numpy()
    exp = expl.explain_instance(
        data_row=x_row,
        predict_fn=rf.predict_proba,
        num_features=min(6, X_test.shape[1]),
        num_samples=3000
    )
    lime_df = pd.DataFrame(exp.as_list(), columns=["Local feature", "Weight"])
    lime_df.to_csv(f"{OUT_DIR}/lime_local_weights_{k+1}.csv", index=False)

    fig, ax = plt.subplots(figsize=(9, 4.2))
    ax.axis("off")
    tb = ax.table(cellText=lime_df.values, colLabels=lime_df.columns, loc="center")
    tb.scale(1, 1.25)
    plt.title("Local Explanation – Example Student Attempt (LIME Weights)")
    plt.tight_layout()
    plt.savefig(f"{OUT_DIR}/lime_local_table_{k+1}.png", dpi=300, bbox_inches="tight")
    plt.savefig(f"{OUT_DIR}/lime_local_table_{k+1}.pdf", dpi=300, bbox_inches="tight")
    plt.show()

    with open(f"{OUT_DIR}/lime_local_table_{k+1}.tex", "w") as f:
        f.write(lime_df.to_latex(index=False, float_format=lambda v: f"{v:.3f}"))

print(" LIME tables saved.")

import zipfile, os, datetime
zip_path = f"xai_outputs_{datetime.datetime.now().strftime('%Y%m%d_%H%M%S')}.zip"
with zipfile.ZipFile(zip_path, "w", zipfile.ZIP_DEFLATED) as zf:
    for root, _, files in os.walk(OUT_DIR):
        for fn in files:
            full = os.path.join(root, fn)
            arc = os.path.relpath(full, OUT_DIR)
            zf.write(full, arcname=f"xai_outputs/{arc}")
print(" ZIP ready:", zip_path)

```

References

1. Mousavinasab, E.; Zarifsanaiey, N.; Kalhori, S.R.N.; Rakhshan, M.; Keikha, L.; Ghazi Saeedi, M. Intelligent Tutoring Systems: A Systematic Review of Characteristics, Applications, and Evaluation Methods. *Interact. Learn. Environ.* **2021**, *29*, 142–163. [\[CrossRef\]](#)

2. Deeksha, S.M. Database Management System and Types of Build Architecture. *Math. Stat. Eng. Appl.* **2022**, *71*, 1380–1388. Available online: <https://www.philstat.org/index.php/MSEA/article/view/630> (accessed on 24 August 2025).
3. Hingorani, K.; Gittens, D.; Edwards, N. Reinforcing database concepts by using entity relationships diagrams (erd) and normalization together for designing robust databases. *Issues Inf. Syst.* **2017**, *18*, 148–155. [\[CrossRef\]](#)
4. Mitrovic, A. An Intelligent SQL Tutor on the Web. *Int. J. Artif. Intell. Ed.* **2003**, *13*, 173–197. [\[CrossRef\]](#)
5. Freeman, J.; Zachary, W. Intelligent Tutoring for Team Training: Lessons Learned from US Military Research. In *Research on Managing Groups and Teams*; Johnston, J., Sottolare, R., Sinatra, A.M., Shawn Burke, C., Eds.; Emerald Publishing Limited: Leeds, UK, 2018; Volume 19, pp. 215–245, ISBN 978-1-78754-474-1.
6. Neves, R.M.; Lima, R.M.; Mesquita, D. Teacher Competences for Active Learning in Engineering Education. *Sustainability* **2021**, *13*, 9231. [\[CrossRef\]](#)
7. Forcael, E.; Garcés, G.; Bastías, E.; Friz, M. Theory of Teaching Techniques Used in Civil Engineering Programs. *J. Prof. Issues Eng. Educ. Pract.* **2019**, *145*, 02518008. [\[CrossRef\]](#)
8. Li, Q.; Chen, Y.-L. Entity-Relationship Diagram. In *Modeling and Analysis of Enterprise and Information Systems*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 125–139, ISBN 978-3-540-89555-8.
9. Yang, L. The Effect of MySQL Workbench in Teaching Entity-Relationship Diagram (ERD) to Relational Schema Mapping. *Int. J. Mod. Educ. Comput. Sci.* **2016**, *8*, 1–12. [\[CrossRef\]](#)
10. Cui, W.; Wang, J.; Huang, H.; Wang, Y.; Lin, C.Y.; Zhang, H.; Zhang, D. IEEE VIS 2021 Virtual: A Mixed-Initiative Approach to Reusing Infographic Charts. *IEEE Trans. Vis. Comput. Graph.* **2021**, *28*, 173–183. Available online: https://virtual-staging.ieeevis.org/year/2021/paper_v-full-1637.html (accessed on 24 August 2025). [\[CrossRef\]](#)
11. Pister, A.; Buono, P.; Fekete, J.-D.; Plaisant, C.; Valdivia, P. Integrating Prior Knowledge in Mixed Initiative Social Network Clustering. *IEEE Trans. Vis. Comput. Graph.* **2020**, *27*, 1775–1785. [\[CrossRef\]](#)
12. Çelikyürek, H.; Karakuş, K.; Aygün, T.; Taş, A. Database Usage and Its Importance in Livestock. *Manas J. Agric. Vet. Life Sci.* **2019**, *9*, 117–121.
13. Watt, A.; Eng, N. *Database Design*, 2nd ed.; The BC Open Textbook Project: Victoria, BC, Canada, 2014.
14. Rashkovits, R.; Lavy, I. Mapping Common Errors in Entity Relationship Diagram Design of Novice Designers. *Int. J. Database Manag. Syst.* **2021**, *13*, 1–19. [\[CrossRef\]](#)
15. Tavana, M. Enterprise Information Systems and the Digitalization of Business Functions. In *Advances in Business Information Systems and Analytics*; IGI Global: Palmdale, PA, USA, 2017; ISBN 978-1-5225-2382-6.
16. Rashkovits, R.; Lavy, I. Students’ Difficulties in Identifying the Use of Ternary Relationships in Data Modeling. *Int. J. Inf. Commun. Technol. Educ.* **2020**, *16*, 47–58. [\[CrossRef\]](#)
17. Saranya, A.; Subhashini, R. A Systematic Review of Explainable Artificial Intelligence Models and Applications: Recent Developments and Future Trends. *Decis. Anal. J.* **2023**, *7*, 100230. [\[CrossRef\]](#)
18. AlShaikh, F.; Hewahi, N. AI and Machine Learning Techniques in the Development of Intelligent Tutoring System: A Review. In *Proceedings of the 2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT)*, Zallaq, Bahrain, 29–30 September 2021; IEEE: New York, NY, USA, 2021; pp. 403–410.
19. Gilbert, S.B.; Dorneich, M.C.; Walton, J.; Winer, E. Five Lenses on Team Tutor Challenges: A Multidisciplinary Approach. In *Research on Managing Groups and Teams*; Johnston, J., Sottolare, R., Sinatra, A.M., Shawn Burke, C., Eds.; Emerald Publishing Limited: Leeds, UK, 2018; Volume 19, pp. 247–277, ISBN 978-1-78754-474-1.
20. Clancey, W.J.; Hoffman, R.R. Methods and Standards for Research on Explainable Artificial Intelligence: Lessons from Intelligent Tutoring Systems. *Appl. AI Lett.* **2021**, *2*, e53. [\[CrossRef\]](#)
21. Khosravi, H.; Shum, S.B.; Chen, G.; Conati, C.; Tsai, Y.-S.; Kay, J.; Knight, S.; Martinez-Maldonado, R.; Sadiq, S.; Gašević, D. Explainable Artificial Intelligence in Education. *Comput. Educ. Artif. Intell.* **2022**, *3*, 100074. [\[CrossRef\]](#)
22. Zhai, X.; Chu, X.; Chai, C.S.; Jong, M.S.Y.; Istenic, A.; Spector, M.; Liu, J.-B.; Yuan, J.; Li, Y. A Review of Artificial Intelligence (AI) in Education from 2010 to 2020. *Complexity* **2021**, *2021*, 8812542. [\[CrossRef\]](#)
23. Gunning, D.; Aha, D.W. DARPA’s Explainable Artificial Intelligence Program. *AI Mag.* **2019**, *40*, 44–58. [\[CrossRef\]](#)
24. Elmadani, M.; Mitrovic, A.; Weerasinghe, A.; Neshatian, K. Investigating Student Interactions with Tutorial Dialogues in EER-Tutor. *Res. Pract. Technol. Enhanc. Learn.* **2015**, *10*, 16. [\[CrossRef\]](#) [\[PubMed\]](#)
25. Hur, P.; Lee, H.; Bhat, S.; Bosch, N. Using Machine Learning Explainability Methods to Personalize Interventions for Students. In *Proceedings of the International Conference on Educational Data Mining*, Durham, UK, 24–27 July 2022. [\[CrossRef\]](#)
26. Tjoa, E.; Guan, C. A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 4793–4813. [\[CrossRef\]](#) [\[PubMed\]](#)
27. Conati, C.; Barral, O.; Putnam, V.; Rieger, L. Toward Personalized XAI: A Case Study in Intelligent Tutoring Systems. *Artif. Intell.* **2021**, *298*, 103503. [\[CrossRef\]](#)
28. Islam, S.R.; Eberle, W.; Ghafoor, S.K.; Ahmed, M. Explainable Artificial Intelligence Approaches: A Survey. *arXiv* **2021**, arXiv:2101.09429. [\[CrossRef\]](#)

29. Lamy, J.-B.; Sekar, B.; Guezennec, G.; Bouaud, J.; Séroussi, B. Explainable Artificial Intelligence for Breast Cancer: A Visual Case-Based Reasoning Approach. *Artif. Intell. Med.* **2019**, *94*, 42–53. [CrossRef]
30. Farrow, R. The Possibilities and Limits of Explicable Artificial Intelligence (XAI) in Education: A Socio-Technical Perspective. *Learn. Media Technol.* **2023**, *48*, 266–279. [CrossRef]
31. Melo, E.; Silva, I.; Costa, D.G.; Viegas, C.M.D.; Barros, T.M. On the Use of eXplainable Artificial Intelligence to Evaluate School Dropout. *Educ. Sci.* **2022**, *12*, 845. [CrossRef]
32. Hostetter, J.W.; Conati, C.; Yang, X.; Abdelshiheed, M.; Barnes, T.; Chi, M. XAI to Increase the Effectiveness of an Intelligent Pedagogical Agent. In Proceedings of the 23rd ACM International Conference on Intelligent Virtual Agents, Würzburg, Germany, 19 September 2023; ACM: Würzburg, Germany, 2023; pp. 1–9.
33. Fiok, K.; Farahani, F.V.; Karwowski, W.; Ahram, T. Explainable Artificial Intelligence for Education and Training. *J. Def. Model. Simul.* **2022**, *19*, 133–144. [CrossRef]
34. Lane, H.C.; Core, M.G.; Van Lent, M.; Solomon, S.; Gomboc, D. Explainable Artificial Intelligence for Training and Tutoring. In Proceedings of the International Conference on Artificial Intelligence in Education (12th), Amsterdam, The Netherlands, 18–22 July 2005; ResearchGate: Berlin, Germany, 2005.
35. Sharma, P. Smart Education Using Explainable Artificial Intelligence (XAI). In *Artificial Intelligence and Machine Learning for Smart Community*; CRC Press: Boca Raton, FL, USA, 2023; pp. 101–110, ISBN 978-1-003-40950-2.
36. Wasilewski, A.; Kolaczek, G. One Size Does Not Fit All: Multivariant User Interface Personalization in E-Commerce. *IEEE Access* **2024**, *12*, 65570–65582. [CrossRef]
37. Andrade, C. The Limitations of Online Surveys. *Indian J. Psychol. Med.* **2020**, *42*, 575–576. [CrossRef]
38. Sabeeh, Z.; Syed Mustapha, S.; Mohamad, R. Healthcare Knowledge Sharing Among a Community of Specialized Physicians. *Cogn. Technol. Work* **2018**, *20*, 105–124. [CrossRef]
39. Vygotsky, L.S. *Mind in Society: The Development of Higher Psychological Processes*; Harvard University Press: Cambridge, MA, USA, 1978.
40. Bloom, B.S. *Taxonomy of Educational Objectives the Classification of Educational Goals Handbook I Cognitive Domain*; Longmans, Green and Co., Ltd.: London, UK, 1956. Available online: <https://www.scirp.org/reference/referencespapers?referenceid=2924447> (accessed on 24 August 2025).
41. WES WES iGPA Calculator—WES.Org. Available online: <https://applications.wes.org/igpa-calculator/> (accessed on 25 August 2025).
42. Palinkas, L.A.; Horwitz, S.M.; Green, C.A.; Wisdom, J.P.; Duan, N.; Hoagwood, K. Purposeful Sampling for Qualitative Data Collection and Analysis in Mixed Method Implementation Research. *Adm. Policy Ment. Health* **2015**, *42*, 533–544. [CrossRef]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.