The Coversheet	
Student Name	Gaurab Shrestha
Student Number	230188112
Word Count / Pages / Duration / Other Limits:	7932 (6779)/37
Attempt Number:	1
Date of Submission:	19th December 2024

I have read and understood the Academic Misconduct statement.	Tick to confirm ☑
I have read and understood the Generative Artificial Intelligence use statement.	Tick to confirm ☑
I am satisfied that I have met the Learning Outcomes of this assignment (Please check the Assignment Brief if you are unsure)	Met ☑

Self-Assessment – If there are particular aspects of your assignment on which you would like feedback, please indicate below. Optional for students
Suggested prompt questions- *How have you developed or progressed you're learning in this work?* *What do you feel is the strongest part of this submission?* *What feedback would you give yourself?* *What part(s) of this assignment are you still unsure about?*

Assessor's Feedback (may be delivered in line with the submission)	
Were the learning outcomes met?	Yes ☑. If not, what was not met:
Assessor's response to the student's submission, request for feedback and / or self-assessment (feedback):	
What specific actions should the student undertake to progress their learning? (feedforward):	
Please take this and other feedback to your next academic tutorial to plan your future work.	

Intelligent Tutoring System
For
Mathematics

(Area Calculation of Different Geometric Shapes)

ABSTRACT

The Intelligent Tutoring System (ITS) for Mathematics focuses on simplifying the calculation of area and surface area for geometric shapes, offering an interactive and personalized learning experience. This system combines a user-friendly Graphical User Interface (GUI), developed with Tkinter, and an ontology-based knowledge representation designed in Protégé. The ITS is structured into two main modules: a Learning Module, which provides dynamic formula-based calculations, real-time area computations, and multimedia resources like YouTube for enhanced understanding; and an Assessment Module, which evaluates students through interactive quizzes, immediate feedback, and corrections to reinforce learning. By leveraging ontology-driven organization and adaptive feedback mechanisms, the ITS ensures a seamless and effective learning process for diverse learners.

This project highlights the successful integration of pedagogical principles with technological innovation to address foundational challenges in mathematics education. Key achievements include a robust system design capable of scaling to other mathematical domains and delivering consistent results through structured knowledge representation. Challenges such as real-time input processing and feedback generation were effectively tackled, ensuring the ITS is both functional and engaging. Future enhancements, such as web-based accessibility, AI-driven personalization, and advanced analytics, could further enrich the system's capabilities. The project underscores the transformative potential of combining technology with user-centric design to make education more inclusive, engaging, and impactful.

Table of Contents

List of Figures

List of Tables

INTRODUCTION

1. Background

Education is a cornerstone of individual and societal development, with mathematics serving as one of its most critical pillars. Mathematics nurtures logical reasoning, critical thinking, and problem-solving skills, which are essential for success across various disciplines and in everyday life (Chakravarthula, 2024). However, mathematics is often perceived as challenging, particularly when it comes to mastering foundational concepts such as calculating the area of geometric shapes. These fundamental concepts—though seemingly simple—form the basis for more advanced topics, including geometry, algebra, trigonometry, and calculus. A strong grasp of these principles is vital to ensure academic progression and a deeper understanding of mathematical problem-solving (Liljedahl, et al., 2016).

The traditional classroom model, while integral to education, frequently struggles to address the diverse learning needs of students. Constraints such as limited time, resources, and the varying paces at which students learn make it difficult for teachers to provide personalized attention. As mathematics builds sequentially on core concepts, any gaps in foundational knowledge can lead to long-term difficulties when tackling more advanced topics (Simms, 2016). This highlights the need for innovative educational approaches that complement conventional teaching methods.

Figure 1: Teaching Methodologies (Binod, 2021)

In recent years, advancements in Artificial Intelligence (AI) have revolutionized education by introducing tools that enable personalized and adaptive learning. AI-powered systems, particularly Intelligent Tutoring Systems (ITS), have emerged as powerful solutions for improving education outcomes (Woolf, 2010). An ITS simulates the role of a human tutor by adapting to the specific needs of individual learners. It monitors a student's progress in real time, identifies areas of difficulty, adjusts the complexity of tasks, and provides targeted feedback. This dynamic and personalized approach to teaching makes learning more engaging, inclusive, and effective.

This project aims to develop an Intelligent Tutoring System specifically designed to teach students how to calculate the area of geometric shapes, including triangles, squares, rectangles, and circles. By leveraging AI techniques, the system will provide an interactive and adaptive learning experience. It will not only deliver instructional content but also engage students through real-time feedback, tailored problem sets, and progressive challenges. The ITS will ensure students gain a solid understanding of area calculations, thereby strengthening their mathematical foundation. By bridging the gap between traditional classroom instruction and the need for individualized learning, this system demonstrates the potential of AI to address persistent challenges in mathematics education. The project underscores how technology can foster deeper understanding, promote student confidence, and equip learners with the skills necessary for academic success and real-world applications.

2. Problem Statement

Mathematics plays a critical role in developing essential skills such as critical thinking and problem-solving, making it a foundational subject in education. However, many students face significant challenges in mastering basic mathematical concepts, such as calculating the area of geometric shapes. These difficulties often arise from a combination of factors:

1. **Lack of Personalization:** Traditional classroom settings adopt a one-size-fits-all approach, which fails to cater to the diverse learning speeds and styles of individual students. Teachers may not have the resources or time to provide the personalized attention that struggling learners need.
2. **Knowledge Gaps**: Students who move on to advanced topics without a solid understanding of foundational concepts may develop gaps in their knowledge, which can hinder their overall academic progress and confidence.
3. **Limited Feedback:** Conventional teaching methods often provide delayed or generic feedback on errors, leading to repeated mistakes and diminishing students' confidence in problem-solving.
4. **Engagement Issues:** Many students find traditional methods of teaching geometry unengaging, which reduces their interest and motivation to learn.

These challenges are particularly significant in geometry, where a strong understanding of basic principles, such as calculating areas, is crucial for success in advanced fields like physics, engineering, architecture, and computer science. While existing digital learning platforms aim to address some of these issues, they often lack true interactivity and adaptability. These platforms may provide practice problems but fail to deliver the real-time feedback, dynamic adjustments, and personalized guidance that emulate a human tutor. This project aims to bridge these gaps by developing an Intelligent Tutoring System (ITS) tailored to teaching the calculation of areas for fundamental geometric shapes such as triangles and squares. The system will:

1. Simulate the role of a knowledgeable, dedicated tutor available at all times.
2. Provide personalized, real-time feedback to correct errors and reinforce learning.
3. Dynamically adapt to the learner's pace, understanding, and progress.
4. Engage students through interactive, adaptive learning experiences.

By addressing the limitations of traditional teaching and existing digital tools, this ITS seeks to create a scalable and effective solution that ensures every learner has the opportunity to build a strong foundation in mathematics.

3. Algorithms

To create an ITS capable of addressing these challenges, several advanced AI techniques can be employed. Each technique contributes to the system's ability to deliver a personalized, engaging, and effective learning experience:

a. Rule-Based Reasoning:

Rule Based Reasoning is ideal for mathematical tasks, where solutions follow deterministic rules (Chhowdhary, 2020). This method involves predefined logical rules for calculating the area of shapes. For instance, if a student is solving for the area of a triangle, the system would guide them to identify the base and height, apply the formula (Area = ½ × base × height), and compute the result. Rule-based systems ensure precision, consistency, and transparency, making them well-suited for teaching fixed mathematical concepts.

b. Natural Language Processing (NLP):

NLP enables the system to interpret and respond to user queries in plain language, enhancing user interaction by allowing students to ask questions and receive answers in a natural, conversational manner (Woolf, 2009). For example, if a student asks, "Why do we multiply by ½ in the area of a triangle?" the system can provide an intuitive explanation. NLP enhances accessibility and engagement by allowing students to interact with the system conversationally, similar to how they would with a human tutor.

c. Reinforcement Learning:

Reinforcement learning enables the ITS to adapt dynamically to a student’s learning progress. By analyzing performance data, the system can adjust the difficulty level of problems, offer hints when need, and reward correct answers with encouraging feedback. This approach personalizes the learning experience, ensuring that students remain challenged but not overwhelmed.

d. Knowledge Representation Frameworks:

This involves structuring mathematical concepts and their relationships in a hierarchical model. For example, "area of a triangle" might be linked to "geometry basics," "multiplication," and "formula application." Knowledge representation ensures that the ITS delivers content logically and progressively, helping students build a solid conceptual foundation.

e. Feedback Mechanism:

Immediate, detailed feedback is provided for each student response. If a student makes an error, the system explains the mistake, highlights the correct steps, and encourages retrying. This fosters active learning and helps students develop a deeper understanding of the material.

4. Rationale

The rationale for developing an Intelligent Tutoring System (ITS) to teach the calculation of areas of geometric shapes lies in addressing key challenges in mathematics education. Foundational concepts like these are essential for progressing to advanced topics, yet many students struggle due to a lack of personalized guidance, delayed feedback, and limited engagement in traditional classrooms. The ITS leverages AI techniques to create an adaptive learning environment that personalizes instruction, providing real-time feedback and progressively adjusting to each learner’s pace. By incorporating rule-based reasoning, the system ensures clarity in mathematical procedures and can be chosen for its reliability in following deterministic processes, which is essential for teaching fixed mathematical concepts like formulas and procedures, while reinforcement learning allows the ITS to evolve dynamically based on student performance which ensures the ITS adapts to individual learners (Wang, et al., 2009). This approach bridges the gap between traditional teaching and the need for individualized support, making education more effective and accessible.

Moreover, the ITS incorporates Natural Language Processing (NLP) to facilitate intuitive interaction, allowing students to ask questions and receive detailed explanations in natural language, enhancing their engagement (Litman, 2016). The use of hierarchical knowledge representation ensures that concepts are introduced logically, promoting deeper understanding and build a strong conceptual foundation. Unlike existing e-learning tools, this ITS is designed to adapt dynamically to the student’s needs, fostering critical thinking and self-directed learning. The choice of these AI techniques reflects a commitment to delivering a scalable, impactful, and student-centered solution that addresses the challenges of modern education while empowering learners to master foundational skills.

PROJECT PLAN

1. Project Overview

The project focuses on designing and developing an Intelligent Tutoring System (ITS) specifically tailored for teaching and assisting students with the calculation of the area of geometric shapes. Geometric shapes like circles, triangles, rectangles, and other polygons are fundamental concepts in mathematics, yet students often struggle to grasp the formulas, visualize the concepts, and apply them effectively in problem-solving.

The **primary objective** of the project is to create an ITS for Mathematics (Focusing on area calculation of different geometric shapes) that is:

- **Interactive and User-Friendly –** Students will receive step-by-step guidance and explanations through an engaging and intuitive interface.
- **Accurate and Comprehensive –** The system will ensure all relevant mathematical content (definitions, formulas, examples, and practice problems) is clearly presented and verified for accuracy.
- **Adaptive to Student Needs –** The ITS will analyze student progress and provide personalized feedback, allowing students to improve their understanding.

This ITS aims to bridge the gap between theory and application by integrating explanations with hands-on practice in an efficient, organized, and technologically advanced manner. The project plan ensures structured execution, well-defined tasks, and strong adaptability to achieve this goal within the set timeframe.

2. Roles and Responsibilities

For a project, like building an intelligent tutoring system, there are different professional roles associated who have different responsibilities that are described below:

A. Project Manager:

Responsibilities: The project manager oversee the entire project, including setting timelines, tracking progress, and ensuring deliverables meet quality standards. The project manager will be the primary point of contact with stakeholders, managing risks, resources, and team communications.

Key Tasks:

- Develop and maintain the project schedule.
- Coordinate team meetings and ensure alignment on goals.
- Manage project scope and facilitate communication between team members and stakeholders.
- Address risks and take corrective action as needed.

B. Lead Developer:

Responsibilities: Lead Developer will lead the technical development of the ITS, including algorithm design, system integration, and software implementation. The Lead Developer will work closely with the Project Manager to ensure technical aspects of the project align with project goals.

Key Tasks:

- Design the system architecture and select appropriate AI techniques.
- Implement rule-based reasoning, NLP, and reinforcement learning algorithms.
- Oversee the integration of data analysis tools and feedback mechanisms.
- Conduct unit testing and ensure software stability.
- Maintain technical documentation.

C. UI/UX Designer:

Responsibilities: UI/UX designer design the user interface and user experience of the ITS, ensuring that it is intuitive, engaging, and accessible for diverse learners.

Key Tasks:

- Develop wireframes and prototypes.
- Conduct user testing to gather feedback.
- Iterate on design based on user feedback.
- Ensure the design aligns with educational goals and is visually appealing.

D. Content Developer/Specialist:

Responsibilities: They create educational content and resources, such as tutorials, problem sets, and multimedia content. The Content Developer will also ensure that the content aligns with curriculum standards and learning objectives.

Key Tasks:

- Develop and write instructional content.
- Integrate multimedia resources (videos, animations) into the system.
- Create problem sets that correspond to different difficulty levels.
- Collaborate with the development team to align content with the system's functionality.

E. Data Scientist:

Responsibilities: Data scientist analyze user data to optimize the learning experience and the adaptive nature of the ITS. The Data Scientist will develop and test machine learning models that drive the system's feedback and personalization mechanisms.

Key Tasks:

- Collect and analyze data to identify patterns in student learning.

- Design and implement machine learning models for feedback and personalization.
- Integrate data analysis tools into the ITS.
- Monitor system performance and refine algorithms as needed.

F. Quality Assurance (QA) Tester:

Responsibilities: QA tester ensure that the ITS functions correctly, identifying bugs, and testing the system to maintain high-quality standards. The QA Tester will work closely with the Lead Developer to resolve issues.

Key Tasks:

- Develop test plans and conduct functional testing.
- Perform usability tests with end-users.
- Document and track bugs, and verify fixes.
- Collaborate with the development team to maintain software quality.

3. Project Milestones and Timelines

For the development of our Intelligent Tutoring System in Mathematics, we are following the given milestones that will guide the project timeline:

Table 1: Project Milestones and Timelines

Milestone	Description	Duration	Team Members
Project Initiation	• Define project scope, objectives, and deliverables. • Assign roles and responsibilities.	1 Week	Project Manager
Requirement Analysis	• Gather and document system requirements. • Finalize project plan and timeline	1 Week	All Team Members
System Design and Architecture	• Design system architecture including AI algorithms, user interface (UI/UX), and feedback mechanisms. • Develop wireframes and prototypes for the user interface. • Complete technical specification documents detailing system requirements.	2 Weeks	Lead Developer, UI/UX Designer
Content Development and Integration	• Develop educational content aligned with the curriculum. • Integrate multimedia resources into the system. • Create practice problems and test scenarios for algorithm training.	2 Weeks	Content Specialist

System Development	• Implement AI algorithms (rule-based reasoning, NLP, reinforcement learning). • Develop core functionality of the system. • Conduct alpha testing to identify issues and perform initial debugging. • Implement data collection and analysis tools.	4 Weeks	Lead Developer
Integration and Testing	• Test and integrate all modules for accuracy. • Refine user interface based on feedback. • Address any remaining issues and prepare for full deployment.	2 Weeks	Tester and QA
Documentation Preparation	• Prepare user manuals and project reports	1 Week	Documentation Lead
Software Deployment and Support	• Deploy the ITS to a wider audience. • Monitor performance and collect user feedback. • Make any final adjustments based on user data and feedback.	1 Week	Project Manage

The total project duration is 14 weeks with the timelines as shown in the below Gantt Chart below:

Table 2: GANTT Chart

Milestone	**Week 1-2**	**Week 3-4**	**Week 5-6**	**Week 7-8**	**Week 9-10**	**Week 11-12**	**Week 13**	**Week 14**
Project Initiation	✓							
Requirement Analysis	✓	✓						
System Design and Architecture		✓	✓					
Content Development and Integration			✓	✓				
System Development				✓	✓	✓		
Integration and Testing					✓	✓		
Documentation Preparation							✓	

Software Deployment and Support								✓

4. Project Management Approaches

For the project, we are adopting a **Hybrid Project Management Approach**, combining elements of the Waterfall and Agile methodologies to balance structured planning with adaptability:

A. Waterfall Approach

Waterfall methodology is a project management approach that arranges tasks and activities in a step-by-step, sequential order (Pressman, 2014). For the tasks such as requirements gathering, content preparation, and system architecture design will follow a sequential Waterfall model. These stages require upfront planning, clear documentation, and thorough reviews to ensure accuracy before moving forward.

Benefits:

- Clear structure and documentation for initial project phases.
- Minimizes errors in mathematical content and system architecture.

B. Agile Approach

Agile methodology is a project management framework that divides projects into iterative phases, often referred to as sprints (Laoyan, 2024). During development, testing, and review phases, an iterative Agile approach will be used. The ITS will be built in small, incremental sprints, allowing frequent testing, improvements, and integration of feedback. Weekly stand-up meetings will help monitor progress and address challenges.

Benefits:

- Flexibility to adapt to unexpected issues or improvements.
- Continuous evaluation ensures early detection and correction of errors.

C. Project Monitoring Tools

A Gantt Chart will be used to visualize the project timeline, ensuring milestones are well-sequenced and achievable. Progress Reviews will occur weekly to assess completed tasks and adjust future goals where necessary. Task assignments will be tracked using a project management tool such as Trello or Asana for real-time updates.

5. Risk Management Strategies

The project recognizes potential risks and includes strategies to mitigate them which are presented in the following table:

Table 3: Risk Management Strategies

Risk	Impact	Mitigation Strategies
Technical Challenges (coding errors, software failures)	Delayed development timelines.	Conduct regular code reviews and debugging sessions. Seek support from mentors or peers for unresolved issues.
Inaccurate Mathematical Content	Incorrect area calculations, reducing system credibility.	All formulas and content will be cross-verified by the content specialist and QA analyst.
Team Member Absences/Delays	Increased workload for remaining team members.	Redistribute pending tasks to other team members as per their availability. Maintain buffer time in the timeline.
Limited Testing Time	Undetected errors or bugs in the final product.	Allocate two full weeks for testing and bug fixes to ensure quality assurance.
Unclear User Experience	Poor student engagement with the ITS.	Conduct usability testing early during the design phase to gather feedback and make improvements.

The combination of risk identification, mitigation strategies, and continuous monitoring ensures that potential problems are addressed promptly, minimizing disruptions.

6. Contingency Plans

To address unforeseen delays or issues, the following contingency plans have been put in place:

a. Buffer Weeks:

- One additional week has been reserved in the timeline for each critical phase (e.g., testing, development).
- This time can be utilized if tasks are delayed or require additional refinements.

b. Task Redistribution:

- If a team member is unavailable or falls behind, their tasks will be reassigned to others with similar skills to ensure continuity.
- For instance, if the content specialist faces delays, the project manager or documentation lead can assist in verifying and finalizing content.

c. Scalable Scope:

- While the primary focus is on geometric shapes like triangles, rectangles, and circles, more complex shapes (e.g., polygons) can be added in a later phase if time permits.

- This ensures the core objectives are completed first, while optional features can be deferred without affecting the overall system quality.

d. Regular Monitoring and Adaptability:

- Weekly meetings will review project progress, identify issues, and adjust schedules if required.
- Real-time updates through tools like Trello or Asana will keep all members informed.

7. Forward Planning and Adaptability

The project demonstrates strong forward planning and adaptability through the following key strategies:

a. Clear Milestone Definition:

- Each milestone is achievable, logically sequenced, and has a specific timeline.
- Progress will be evaluated at each milestone to ensure the project stays on track.

b. Flexible Approach:

- Combining Waterfall and Agile methods allows for structured content planning with iterative development and testing.

c. Task Accountability:

- Tasks are clearly assigned to team members based on their strengths, ensuring everyone knows their responsibilities.
- Weekly progress reviews allow for quick adjustments in case of challenges.

d. Buffer Time:

- Buffer weeks and contingency plans demonstrate preparedness for unexpected delays, ensuring timely project completion.

LITERATRE REVIEW

This section provides the review of existing Intelligent Tutoring Systems, critique about their structure, mechanism ad knowledge representation in the domain we choose i.e., Mathematics.

1. Overview of Intelligent Tutoring System

Intelligent Tutoring Systems (ITS) are advanced computer-based learning environments that leverage artificial intelligence (AI) to replicate the personalized guidance of a human tutor. These systems aim to enhance education by offering adaptive, tailored instruction that meets the unique needs of individual learners. By integrating sophisticated techniques such as user modeling, rule-based reasoning, semantic knowledge representation, and adaptive feedback mechanisms, ITS provide real-time feedback, identify gaps in understanding, and dynamically adjust educational content to suit each student's progress and learning style.

Particularly impactful in mathematics education, ITS address challenges like varying levels of proficiency, math anxiety, and the scarcity of skilled tutors. Systems such as ActiveMath and ALEKS exemplify how ITS can adapt instructional methods to individual learners, incorporating interactive problem-solving activities and feedback loops to foster engagement and efficiency. By creating an interactive and supportive learning environment, ITS empower students to overcome obstacles, improve comprehension, and achieve their educational goals more effectively than traditional methods alone.

2. Studies on Existing Intelligent Tutoring Systems and Their Critiques

The Intelligent Tutoring System for Mathematics made by (Siekmann & Erica, 2004), ActiveMath was a web-based intelligent tutoring system designed to support personalized, adaptive, and exploratory mathematics learning through AI techniques like user modeling, rule-based reasoning, and semantic knowledge representation. It used a modular client-server architecture, integrating components like a course generator, student model, and external tools such as computer algebra systems to provide interactive problem-solving and tailored feedback. The system employed OMDoc, a semantic XML language, to ensure content reusability, interoperability, and machine-human interpretability, enhancing its long-term applicability. While the separation of pedagogical, user, and content knowledge supports flexibility and scalability, reliance on rule-based systems like Jess and complex standards like OMDoc can limit adaptability and ease of use. Moreover, challenges like real-time interaction bottlenecks and addressing non-mathematical content suggest areas for improvement, though its robust design and adherence to open standards make it a valuable tool in adaptive learning environments.

The paper by (Huang, et al., 2016) explored the impact of the ALEKS Intelligent Tutoring System (ITS) on reducing math achievement gaps among 6th-grade students in after-school programs. It highlighted how ITS effectively equalizes performance across racial, gender, and socioeconomic divides by providing personalized, bias-free feedback and precise knowledge assessments. The modular design of ALEKS, based on Knowledge Space Theory, tailored content to each student's readiness, surpassing traditional teacher-led classes in ensuring consistent outcomes. Critiquing its structure, the reliance on individual interactions ensured inclusivity but may limit scalability in larger groups. Mechanistically, ALEKS exceled with adaptive learning and feedback but lacks integration with collaborative learning features found in classrooms. Its knowledge representation, while precise and detailed, may not address broader conceptual integration or contextual knowledge as effectively as human-led methods. Overall, ITS like ALEKS demonstrated promise in addressing educational inequities, though complementary strategies could enhance its scalability and holistic educational impact.

The paper from (Arroyo, et al., 2010) explored how integrating math fluency training with the Wayang Outpost Intelligent Tutoring System (ITS) improved standardized math test performance, emphasizing cognitive efficiency in problem-solving. The system combined adaptive tutoring with targeted fluency training to free up working memory, enhancing performance, especially on complex problems. Structurally, its modular design accommodated supplemental features like fluency modules, ensuring adaptability and scalability. Mechanistically, the ITS provided personalized feedback and tracks progress effectively, though the study highlights those longer interventions might yield more significant group-specific benefits. In terms of knowledge representation, the system's design prioritized accuracy and real-time adaptivity, but it could expand fluency concepts beyond basic operations for broader applicability. While effective, the results suggest potential for deeper integration of foundational and advanced mathematical skills within ITS frameworks.

The paper from (Shih, et al., 2023) introduced a dialogue-based intelligent tutoring system (ITS) designed to teach sixth graders multiplication and division of fractions. It used diagnostic teaching principles and block-based matching to analyze student responses and address misconceptions. The ITS offerd adaptive instruction through personalized dialogues, guiding students step-by-step with cognitive conflict strategies, problem simplifications, and visual representations. A quasi-experimental study demonstrated the system's effectiveness, with significant performance gains, especially among lower-performing students. While innovative in integrating detailed diagnostic methods, the system's reliance on structured inputs and limited adaptability for complex problem-solving highlights areas for further development. This structure effectively aligned ITS capabilities with pedagogical principles but lacks flexibility for broader mathematical contexts.

The author from the paper (Pirvulescu, 2020) examined the evolution and application of Intelligent Tutoring Systems (ITS) in mathematics, highlighting their role in addressing math anxiety and enhancing personalized learning. Modern ITS, such as the Practical Algebra Tutor (PAT) and AnimalWatch, integrated AI to provide real-time feedback, adaptive lessons, and emotional support, enabling students to learn at their own pace while reducing anxiety. Critically, ITS rely on structured lessons designed by experts, which limits student-driven exploration compared to more open-ended learning methods like coding. Knowledge representation is often curriculum-focused, providing procedural guidance but insufficient conceptual depth or flexibility for diverse problem-solving approaches. While impactful in boosting confidence and academic outcomes, especially for underperforming or anxious students, ITS development faces challenges such as data privacy concerns and infrastructure limitations, indicating a need for more inclusive and adaptable frameworks.

The paper by (Niño, et al., 2023) provided a systematic review of trends in Intelligent Tutoring Systems (ITS) within mathematics education, focusing on secondary schools and higher education institutions (HEIs). ITS are recognized for their role in enhancing personalized learning, adaptive feedback, and supporting mathematical problem-solving skills. However, the study critiqued ITS structures for their limited long-term impact due to small sample sizes, network issues, and underdeveloped technical frameworks. The mechanisms, while beneficial in providing individualized feedback and assessment, often lack flexibility in addressing diverse user profiles. Knowledge representation in ITS was heavily reliant on procedural rather than conceptual frameworks, restricting broader applicability. The paper called for longitudinal studies and more robust adaptive processes to optimize ITS efficacy.

3. Addressing Gaps in Current Systems

To maximize the potential of Intelligent Tutoring Systems (ITS) and broaden their applicability, especially in mathematics education, future developments must address key areas for improvement. First, integrating collaborative learning features alongside personalized instruction can bridge the gap between individual and social learning. Incorporating group problem-solving activities, peer interaction, and feedback mechanisms within ITS frameworks can foster a more holistic and interactive learning environment.

Second, expanding the focus from procedural fluency to conceptual knowledge representation is critical. By supporting diverse problem-solving approaches and fostering a deeper understanding of underlying concepts, ITS can enhance learners' abilities to apply knowledge flexibly across varied contexts. This balanced approach will make ITS more effective in addressing the complex needs of students with different learning styles.

Third, leveraging advanced AI techniques to improve scalability, real-time adaptivity, and interaction capabilities will address technical limitations and expand access to ITS. These advancements can ensure that ITS meet the needs of a wider range of learners and educational institutions. Integrating ITS with existing educational frameworks and tools can also provide a seamless experience, reducing dependency on standalone systems and increasing adoption.

Finally, longitudinal studies, as emphasized by Niño et al. (2023), are essential for evaluating the long-term impact of ITS on educational outcomes. User-centered design approaches should also guide future development to ensure that ITS are adaptable to diverse user profiles, cultural contexts, and educational needs. Addressing data privacy concerns and ensuring accessibility will further enhance inclusivity and acceptance.

By focusing on these advancements, ITS can become more effective, inclusive, and sustainable tools, transforming education for a diverse range of learners and fostering better outcomes across varied learning environments.

DEVELOPMENT OF INTELLIGENT TUTORING SYSTEM

1. Domain Description

The Intelligent Tutoring System (ITS) developed for this project focuses on mathematics, specifically targeting the calculation of areas for fundamental geometric shapes such as triangles, squares, rectangles, and circles. Geometry, as a foundational branch of mathematics, plays a critical role in the school curriculum and serves as a prerequisite for understanding more complex concepts like surface areas, volumes, and trigonometry. Despite its importance, many students face challenges in grasping area formulas, their applications, and the logic behind them. These struggles often arise from the abstract nature of geometric concepts, the lack of personalized feedback, limited interactivity in traditional teaching methods, and difficulties in visualizing shapes.

To address these issues, the proposed ITS offers a dynamic, interactive, and personalized learning environment. The system leverages ontology-based knowledge representation to define geometric relationships and formulas, integrating this structured knowledge base with a functional Graphical User Interface (GUI) developed in Python. The ontology ensures precise calculations and explanations, while the user interface allows for seamless interaction, visualization, and immediate feedback.

The ITS is designed to:

- Provide personalized feedback tailored to user inputs.
- Deliver step-by-step explanations for area calculations.
- Enhance conceptual understanding through visual representations of shapes.
- Enable students to learn at their own pace in an interactive environment.
- Correct errors and offer guidance to improve problem-solving skills.

By combining AI-driven ontology development with interactive programming, the ITS creates a novel solution that improves student engagement, conceptual clarity, and mathematical learning outcomes. This system not only helps students overcome the challenges of understanding geometric concepts but also fosters deeper comprehension and confidence in their mathematical abilities.

2. Requirements Analysis for the ITS

The first step in developing the ITS is understanding the requirements of the system and the target learners. As we are going to build Intelligent Tutoring System for Mathematics focusing the area calculation f different geometric shapes, the following requirements were identified.

A. Domain Specific Requirements

- Focus on teaching area calculations for basic geometric shapes:
  - Circle: Requires radius.
  - Rectangle: Requires length and breadth.
  - Square: Requires side length.
  - Triangle: Requires base and height.

- Provide real-time feedback to guide learners in solving problems.
- Include step-by-step explanations to improve understanding.
- Support input validation to ensure only correct dimensions are processed.
- Display visual representations of geometric shapes to enhance conceptual clarity.

B. System Requirements

a. Ontology Development:

- Create a structured ontology that defines shapes, their properties, and area formulas.

b. User Interface (UI):

- Develop an interactive and visually appealing GUI where users can input values, view solutions, and receive feedback.

c. Integration:

- Ensure seamless communication between the ontology and the user interface.

3. Steps in Developing the ITS

For the development of the Intelligent Tutoring System, it is divided into three phases: **Ontology Creation, User Interface Development and System Integration.**

3.1 Ontology Development Using Protégé

An ontology is a collection of concepts and categories within a specific domain or subject area, describing their attributes and the relationships between them (Earley, 2024). The ontology serves as the knowledge base for the ITS, organizing concepts, relationships, and constraints within the domain of area calculations. The ontology was developed using the Protégé Ontology Editor, which allows for the creation of structured domain knowledge in Web Ontology Language (OWL). The following steps were followed to create the ontology using Protégé:

1. Ontology IRI setup:

The first step in ontology creation is to setup the ontology IRI as we like editing the default IRI. The figure below shows the setup of ontology IRI setup:

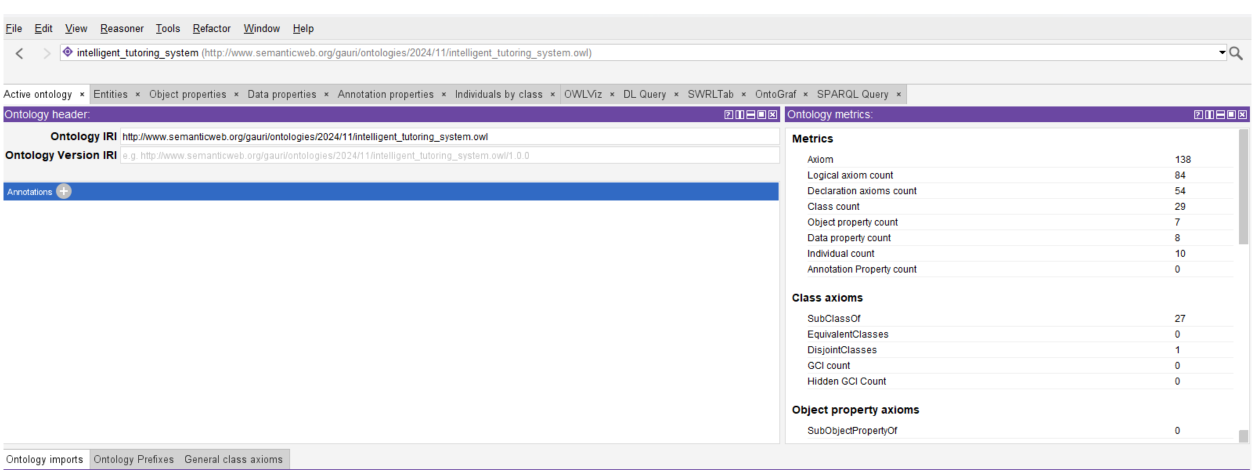

Figure 2: Ontology IRI setup

2. **Class and Subclasses Definition:**
The main classes and subclasses were identified during the requirements analysis of the ITS. Figure below shows the class and subclasses hierarchy for our Intelligent Tutoring System.

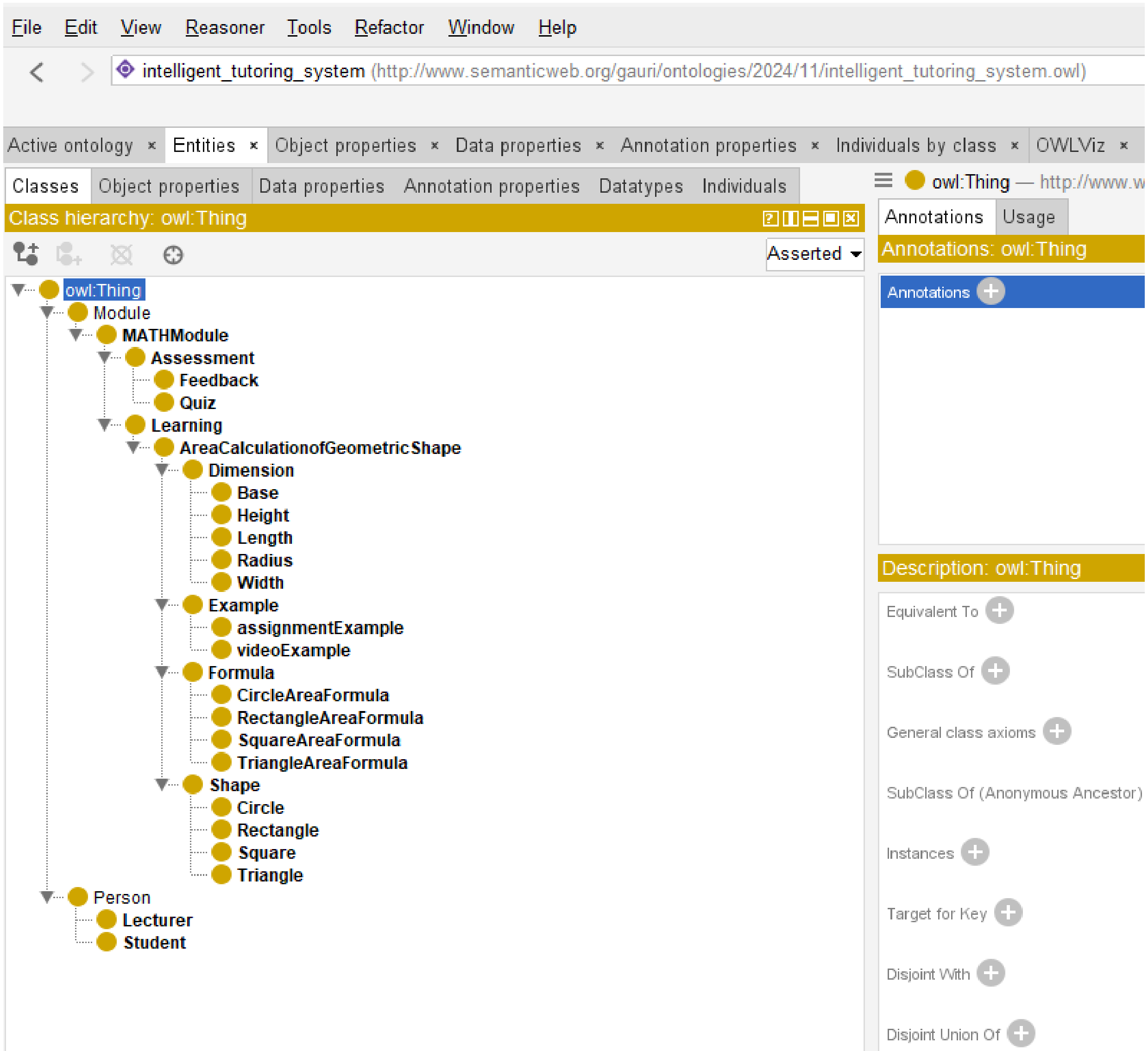

Figure 3: Classes and Subclasses Hierarchy for the ITS

We have two main classes for our ITS: '**Person**' and '**Module**'. 'Person' class have two subclasses '**Lecturer**' and '**Student**' who are the participating person in our ITS. We are more focusing on area calculation in our ITS so, our 'Module' class is a subject class that have '**MATHModule**' subclass which focuses on Mathematics subject that contain many subclasses. The different subclasses under MATHModule are '**Assessment**' and '**Learning**' subclasses which are discussed below:

a. Assessment subclass:

This subclass contains the assessment for the student where there are '**Quiz**' and '**Feedback**' subclasses which contains questions for assessment and feedback for the submission of the assessment.

b. Learning Subclass:

This subclass contains the content for learning which have subclass '**AreaCalculaionofGeometricShape'.** This subclass also contains more subclasses under it.

- **Shape Subclass: Circle, Rectangle, Square, Triangle**
- **Dimension Subclass: Base, Height, Length, Width, Radius**
- **Formula Subclass: CircleAreaFormula, RectangleAreaFormula, SquareAreaFormula**
- **Example Subclass: assignmentExample, videoExample**

3. Property Specification:

We have two properties: **Object Properties** and **Data Properties**.

a. Object Properties: This establishes the relationships between different classes like Shape and Dimension class with '**hasDimension**'. We have different object properties which is shown in below figure:

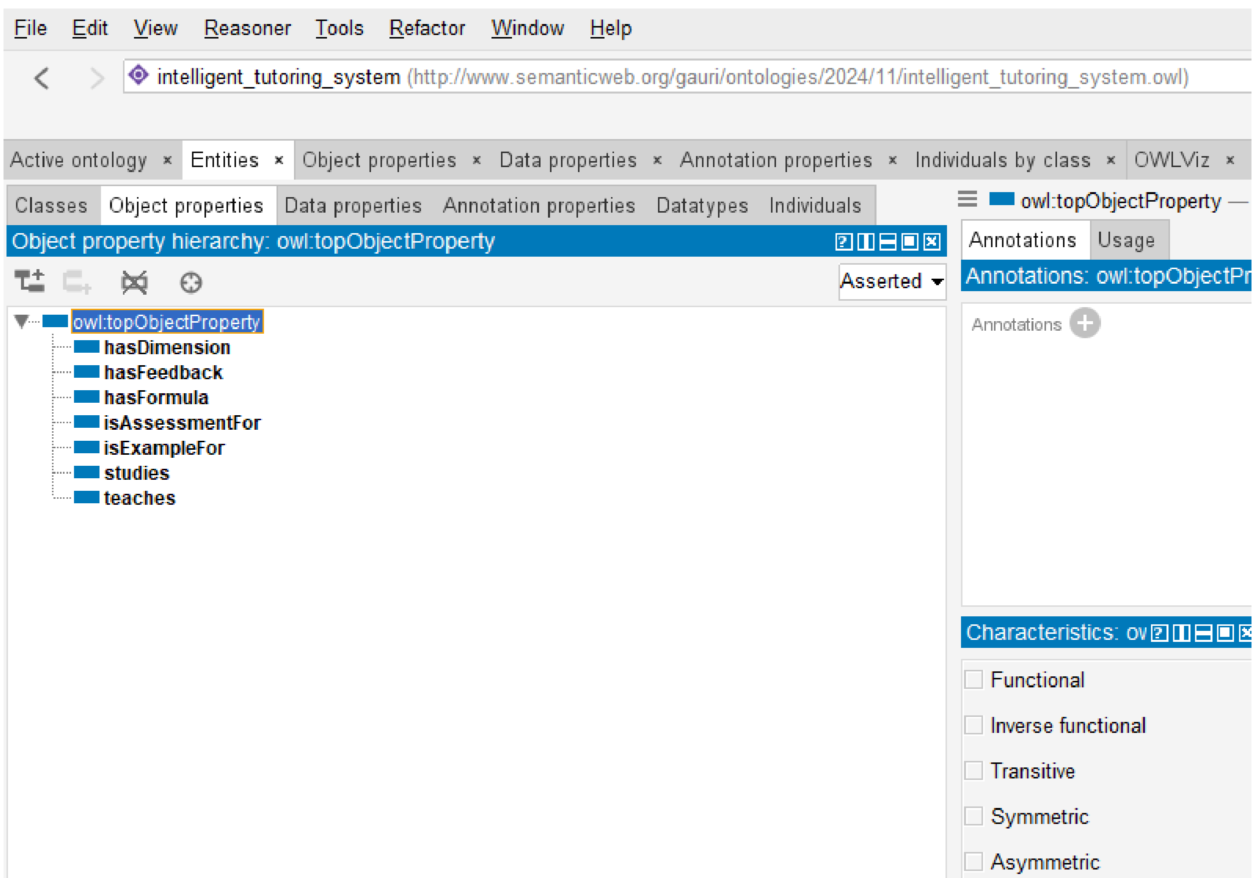

Figure 4: Object Properties for our ITS

b. Data Properties:

Data properties define relationships between individuals (instances of classes) and literal values, such as strings, numbers, dates, or Booleans. Here, one of the data properties define attributes specific to each shape. The following figures shows the data properties of our ITS.

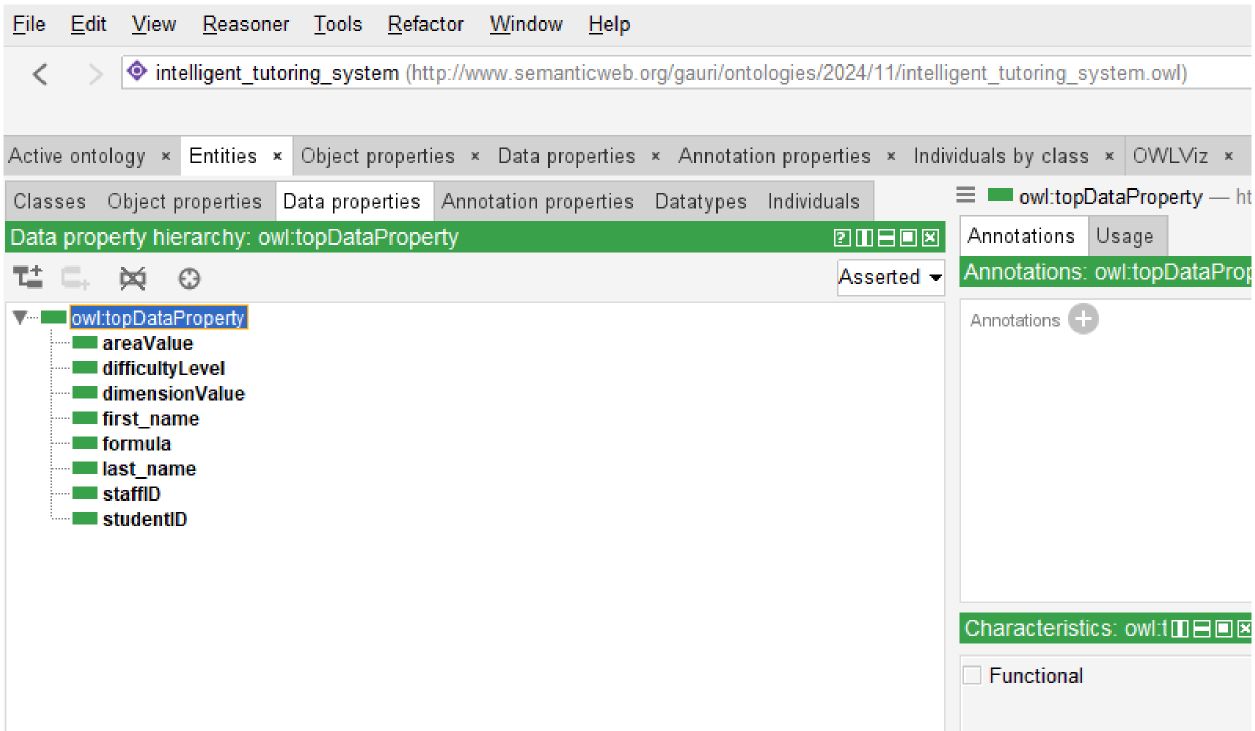

Figure 5: Data Properties of Our ITS

4. Creation of Individuals (Instances):

Individuals populate the ontology classes with example data. For example, Instances like '**Rectangle1**' with **hasDimension = 10** and **hasDimension = 6.**

We have different individuals liked with different classes which is shown in figure below:

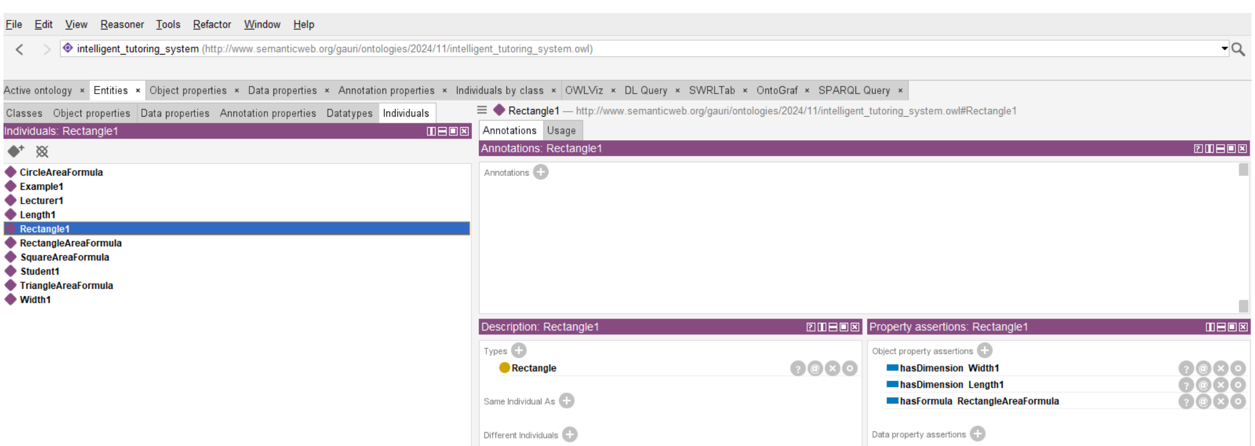

Figure 6: Different Individuals of our Different Classes

5. **Ontology Export:**
We saved the ontology in Web Ontology Language (OWL) format for compatibility with the application.

The following figure shows the final connection of our classes, subclasses, object properties, data properties, individuals which is made with Protégé plugin named ‘**OntoGraf**’.

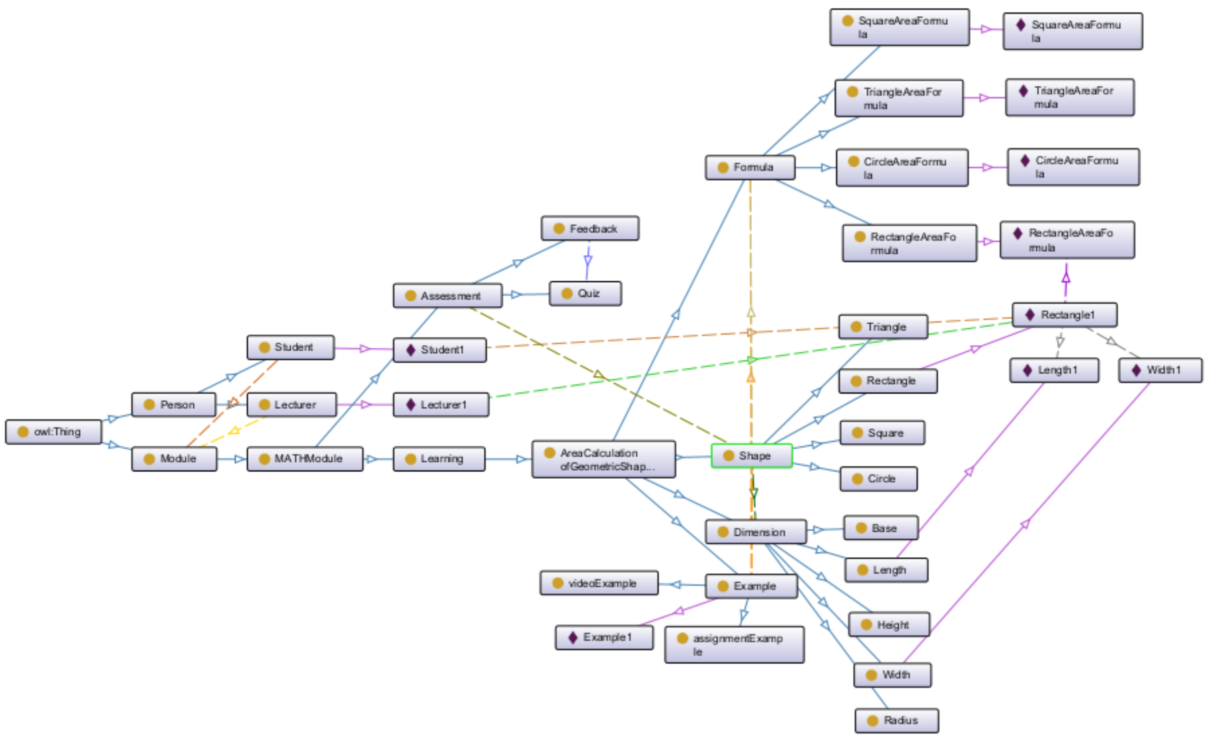

Figure 7: Graph Made with OntoGraf Plugin

3.2 User Interface (UI) Development

The User Interface (UI) serves as the interactive layer between the learner and the ITS. It is developed using Python and its library Tkinter for GUI design. This provides a user-friendly and intuitive platform for students to interact with the ITS and allows seamless querying of the ontology to retrieve formulas, validate inputs and displaying results. The key features of the User Interface of out ITS:

a. **Separate Tab for Learning and Assessment:** There is two different tabs for Learning and Assessment as we have two classes in the ontology.
b. **Shape Selection Menu:** User can select different geometric shapes (circle, square, rectangle, triangle).
c. **Input Fields and Examples:** After selecting the shape, the formulas, YouTube link and input fields are visible to enter the required dimensions for the selected shape (e.g., radius for circles, base and height for triangles).
d. **Real-time Feedback:** If incorrect and invalid values are entered, the system displays an error message and hints for correction.

We created GUI using ‘**tkinter**’ library of Python. We have used YouTube link which will be opened using ‘**webbrowser**’ library. The required libraries to create the GUI for the ITS is shown in the figure below:

```
import tkinter as tk
from tkinter import ttk
import webbrowser
from owlready2 import get_ontology
```

Figure 8: Importing Libraries for the Python Code

The GUI was made using the class “**IntelligentTutoringSystem**” which was initiated with __init__(self, root, ontology_path) method where we have given the title name for GUI, shape geometry for the GUI, loaded the ontology, created the learning and assessment tab, etc.

```
class IntelligentTutoringSystem:
    Codeium: Refactor | Explain | Generate Docstring
    def __init__(self, root, ontology_path):
        self.root = root                [Codeium] Generate docstring for __init__
        self.root.title("Intelligent Tutoring System For Mathematics (Area Calculation of Geometrics Shapes)")
        self.root.geometry("800x600")

        # Load ontology
        self.ontology = get_ontology(ontology_path).load()

        # Create a tab control
        self.tab_control = ttk.Notebook(root)

        # Create tabs
        self.learning_tab = ttk.Frame(self.tab_control)
        self.assessment_tab = ttk.Frame(self.tab_control)

        self.tab_control.add(self.learning_tab, text='Learning')
        self.tab_control.add(self.assessment_tab, text='Assessment')

        self.tab_control.pack(expand=1, fill='both')

        self.create_learning_tab()
        self.create_assessment_tab()

        # Initialize assessment variables
        self.current_level = 1
        self.questions = self.load_questions_from_ontology()
```

Figure 9: Code for Initialization of the GUI

Below figure shows the function to show the formula of any geometrical shapes along with the link for the YouTube video as an example.

```python
def show_formula(self):
    shape = self.shape_var.get()
    self.clear_input_fields()

    if shape == 'Circle':
        self.formula_label.config(text="Area = π * r²")
        self.youtube_link_label.config(text="YouTube Link: Area of Circle")
        self.youtube_link = "https://www.youtube.com/results?search_query=area+of+circle"
        self.create_input_fields(["Enter radius of circle"])
    elif shape == 'Rectangle':
        self.formula_label.config(text="Area = length * breadth")
        self.youtube_link_label.config(text="YouTube Link: Area of Rectangle")
        self.youtube_link = "https://www.youtube.com/results?search_query=area+of+rectangle"
        self.create_input_fields(["Enter length of rectangle", "Enter breadth of rectangle"])
    elif shape == 'Square':
        self.formula_label.config(text="Area = side²")
        self.youtube_link_label.config(text="YouTube Link: Area of Square")
        self.youtube_link = "https://www.youtube.com/results?search_query=area+of+square"
        self.create_input_fields(["Enter side of square"])
    elif shape == 'Triangle':
        self.formula_label.config(text="Area = 0.5 * base * height")
        self.youtube_link_label.config(text="YouTube Link: Area of Triangle")
        self.youtube_link = "https://www.youtube.com/results?search_query=area+of+triangle"
        self.create_input_fields(["Enter base of triangle", "Enter height of triangle"])
    elif shape == 'Cube':
        self.formula_label.config(text="Surface Area = 6 * side²")
        self.youtube_link_label.config(text="YouTube Link: Surface Area of Cube")
        self.youtube_link = "https://www.youtube.com/results?search_query=surface+area+of+cube"
        self.create_input_fields(["Enter side of cube"])
    elif shape == 'Cuboid':
        self.formula_label.config(text="Surface Area = 2 * (length * breadth + breadth * height + height * length)")
        self.youtube_link_label.config(text="YouTube Link: Surface Area of Cuboid")
        self.youtube_link = "https://www.youtube.com/results?search_query=surface+area+of+cuboid"
        self.create_input_fields(["Enter length of cuboid", "Enter breadth of cuboid", "Enter height of cuboid"])
```

Figure 10: Code to Show Formula, YouTube links and Input Fields

To calculate the area of different shapes, we have used the code shown in below figure.

```python
def calculate_area(self):
    shape = self.shape_var.get()
    inputs = [entry.get() for entry in self.input_frame.entries if entry.get()]
    try:
        if shape == 'Circle':
            r = float(inputs[0])
            area = 3.14 * r ** 2
            self.result_label.config(text=f"Area of Circle: {area:.2f} cm²")
        elif shape == 'Rectangle':
            length = float(inputs[0])
            breadth = float(inputs[1])
            area = length * breadth
            self.result_label.config(text=f"Area of Rectangle: {area:.2f} cm²")
        elif shape == 'Square':
            side = float(inputs[0])
            area = side ** 2
            self.result_label.config(text=f"Area of Square: {area:.2f} cm²")
        elif shape == 'Triangle':
            base = float(inputs[0])
            height = float(inputs[1])
            area = 0.5 * base * height
            self.result_label.config(text=f"Area of Triangle: {area:.2f} cm²")
        elif shape == 'Cube':
            side = float(inputs[0])
            surface_area = 6 * side ** 2
            self.result_label.config(text=f"Surface Area of Cube: {surface_area:.2f} cm²")
        elif shape == 'Cuboid':
            length = float(inputs[0])
            breadth = float(inputs[1])
            height = float(inputs[2])
            surface_area = 2 * (length * breadth + breadth * height + height * length)
            self.result_label.config(text=f"Surface Area of Cuboid: {surface_area:.2f} cm²")
    except ValueError:
        self.result_label.config(text="\u26A0\uFE0F Please enter valid numerical values.")
```

Figure 11: Code to Calculate the Area of Geometric Shapes

The complete code can be found in the GitHub repository (Intelligent Tutoring System).

3.3 System Integration and Testing

Integration ensures that the user interface and the ontology communicate seamlessly, providing an efficient and responsive ITS experience. This is achieved using Owlready2, a Python library that allows querying and processing OWL files.

```
from owlready2 import get_ontology
ontology_path = "D:\MSc Classes\SEM-2\AI\intelligent_tutoring_system.owl"
```

Figure 12: Ontology Importing with owlready2 Library

We followed the following steps in Integration of Ontology and User Interface:

1. We loaded the OWL ontology file into the Python environment which is shown in the below figure.

```
# Load ontology
self.ontology = get_ontology(ontology_path).load()
```

Figure 13: Loading Ontology to the Python Code for User Interface

2. Formulas were retrieved and validated inputs based on the shape selected by the user.
3. The user inputs were passed to the ontology for checking against defined constraints.
4. The area calculations, explanations and feedback were displayed in the user interface.

The result is a fully functional system where the interface seamlessly interacts with the ontology to deliver an efficient and interactive learning experience. To ensure functionality, usability and performance, the following testing phases were conducted:

1. **Functional Testing:**
   Functional testing verifies all the functionalities, such as shape selection, input validation, area calculation and feedback mechanisms.
2. **Usability Testing:**
   Usability testing evaluate the user-friendliness and visual appeal of the GUI.
3. **Performance Testing:**
   This ensures seamless communication between the interface and ontology with minimal response time.

3.4 Final Output

After the complete ontology development, user interface development, system integration and testing, our final graphical user interface for the system is presented in the figure below. We have given title of the GUI "Intelligent Tutoring System for Mathematics (Area Calculation of Geometric Shapes). We have two tabs at the top left corner: Learning and Assessment. In the learning tab, we can select the different geometric shapes.

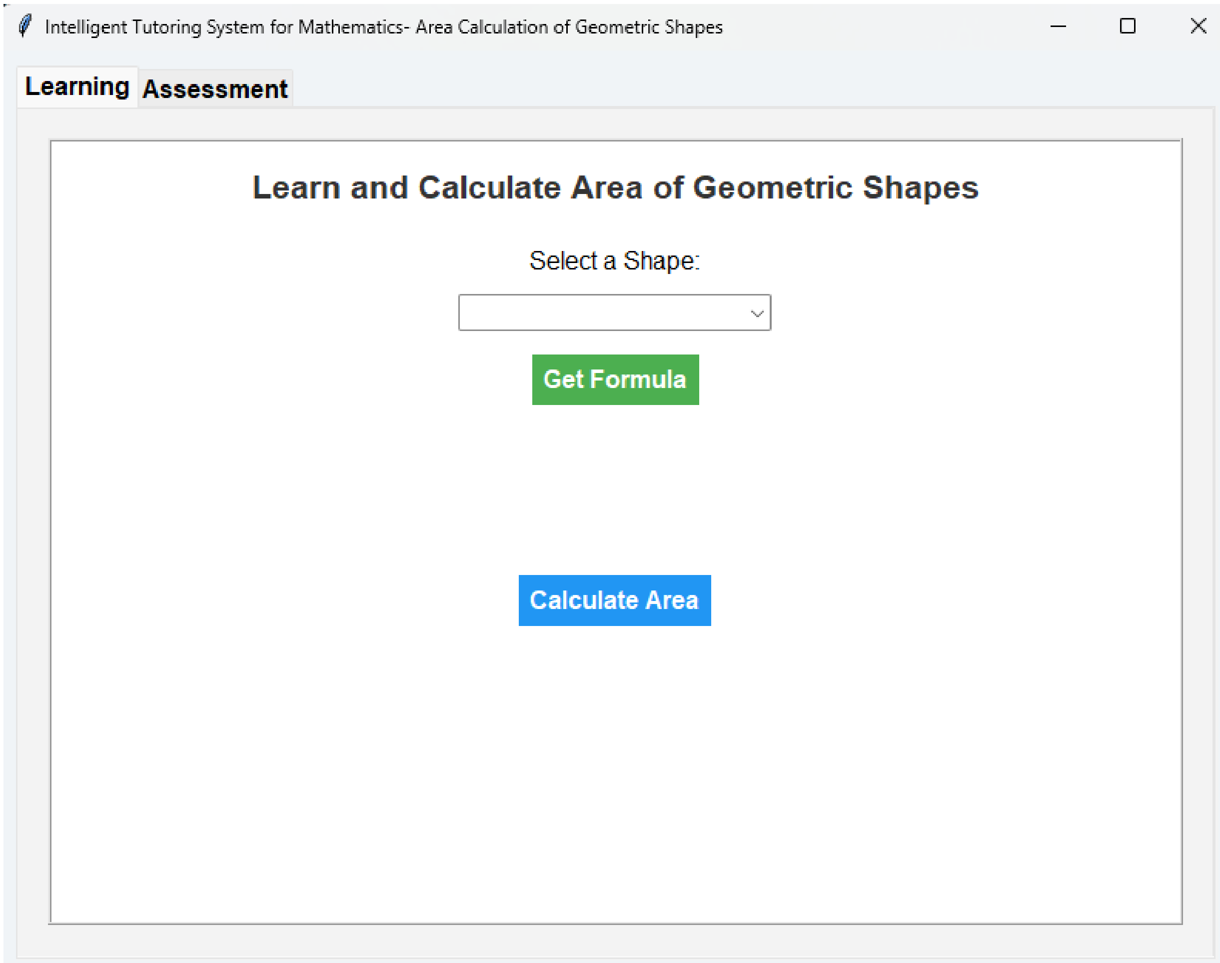

Figure 14: Landing Page of our GUI for ITS

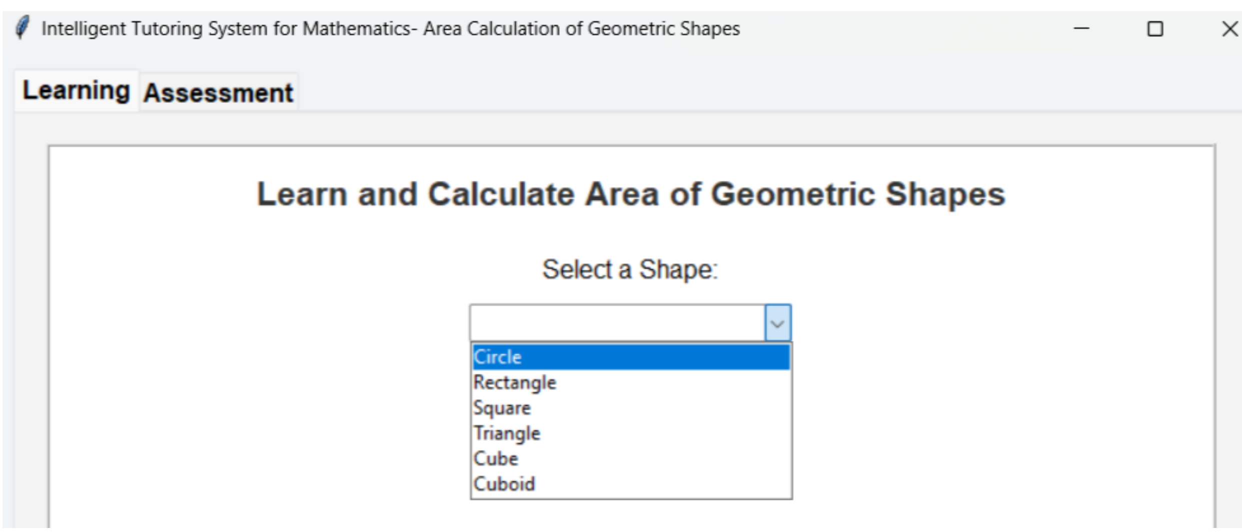

Figure 15: List of Geometric Shapes in Our ITS

In the system, we have included different basic geometrical shapes that can be accessed from the dropdown menu which is shown in above figure 15.

We selected and rectangle from the dropdown menu and clicked the 'Get Formula' button which will give us the result as shown in figure 16. The results include the formula to calculate the area of rectangle, YouTube link to see the example of calculating the area of rectangle, and two input fields. After inputting the values and click 'Calculate Area' button, we got the area of rectangle that is shown in the figure below.

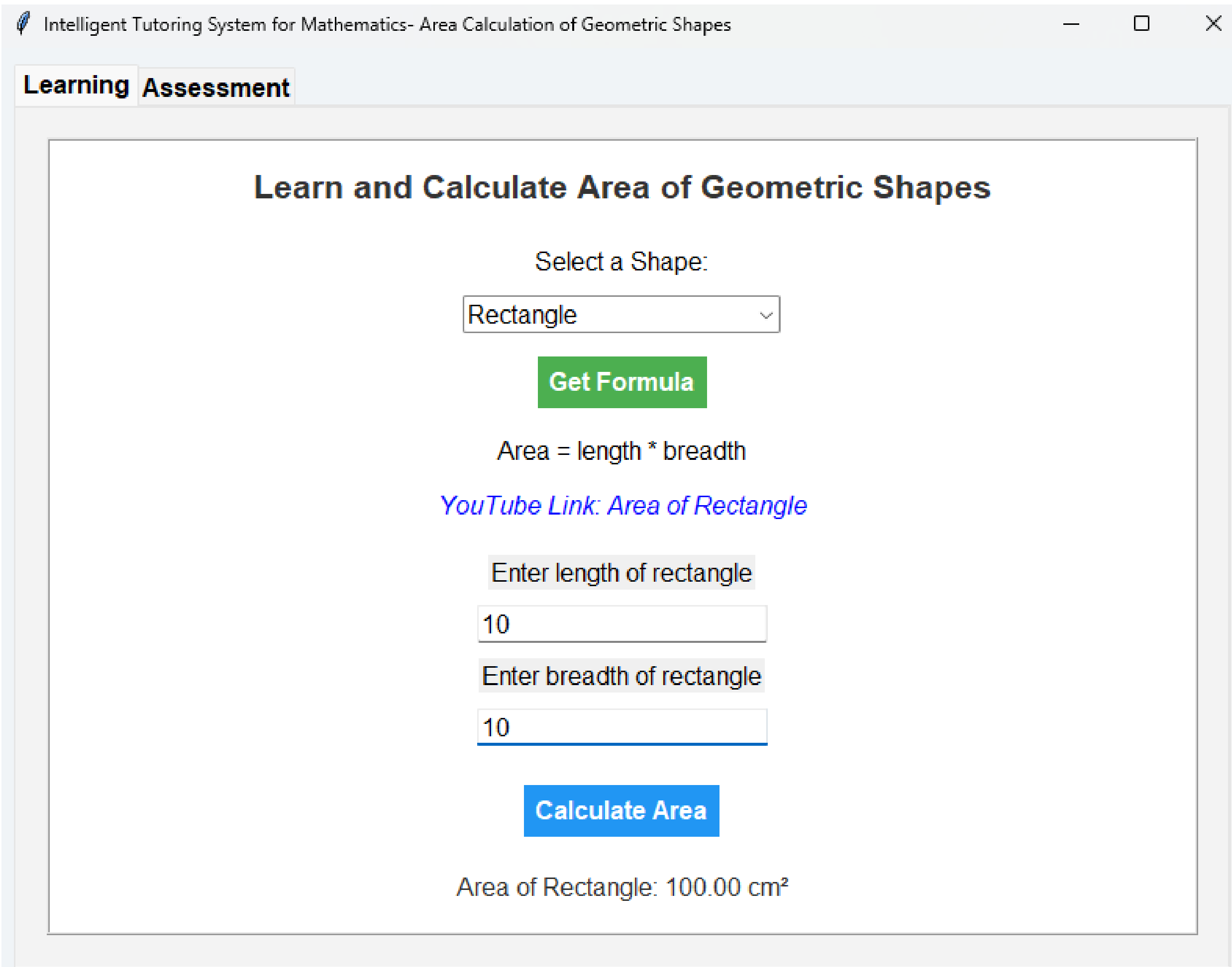

Figure 16: Calculating the Area of Rectangle

In the assessment tab, we have different questions related to calculating the area of geometric shapes. The GUI is presented in the below figure.

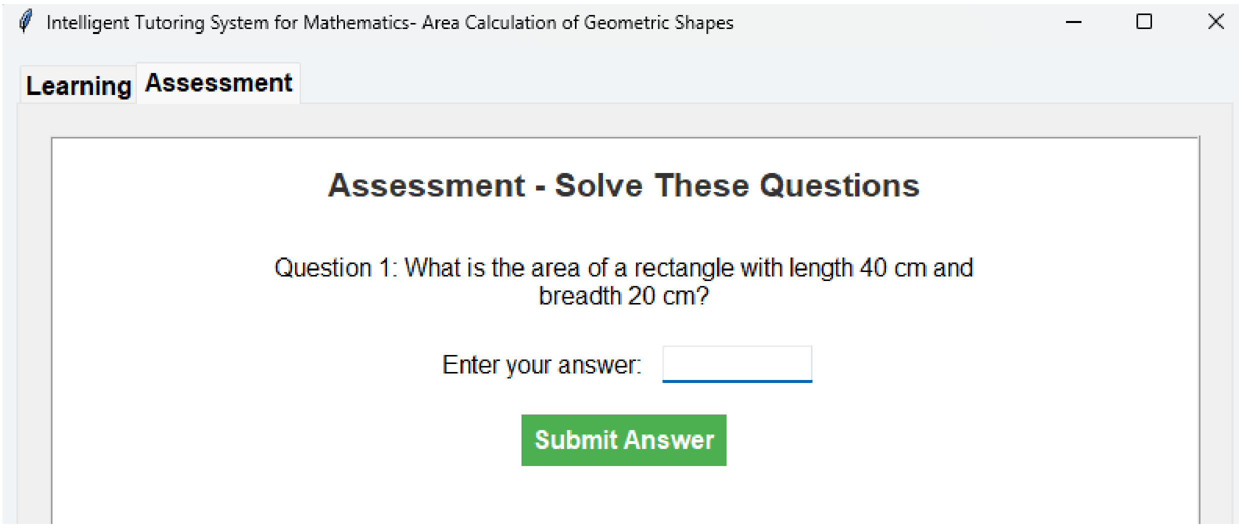

Figure 17: Assessment Tab of Our ITS

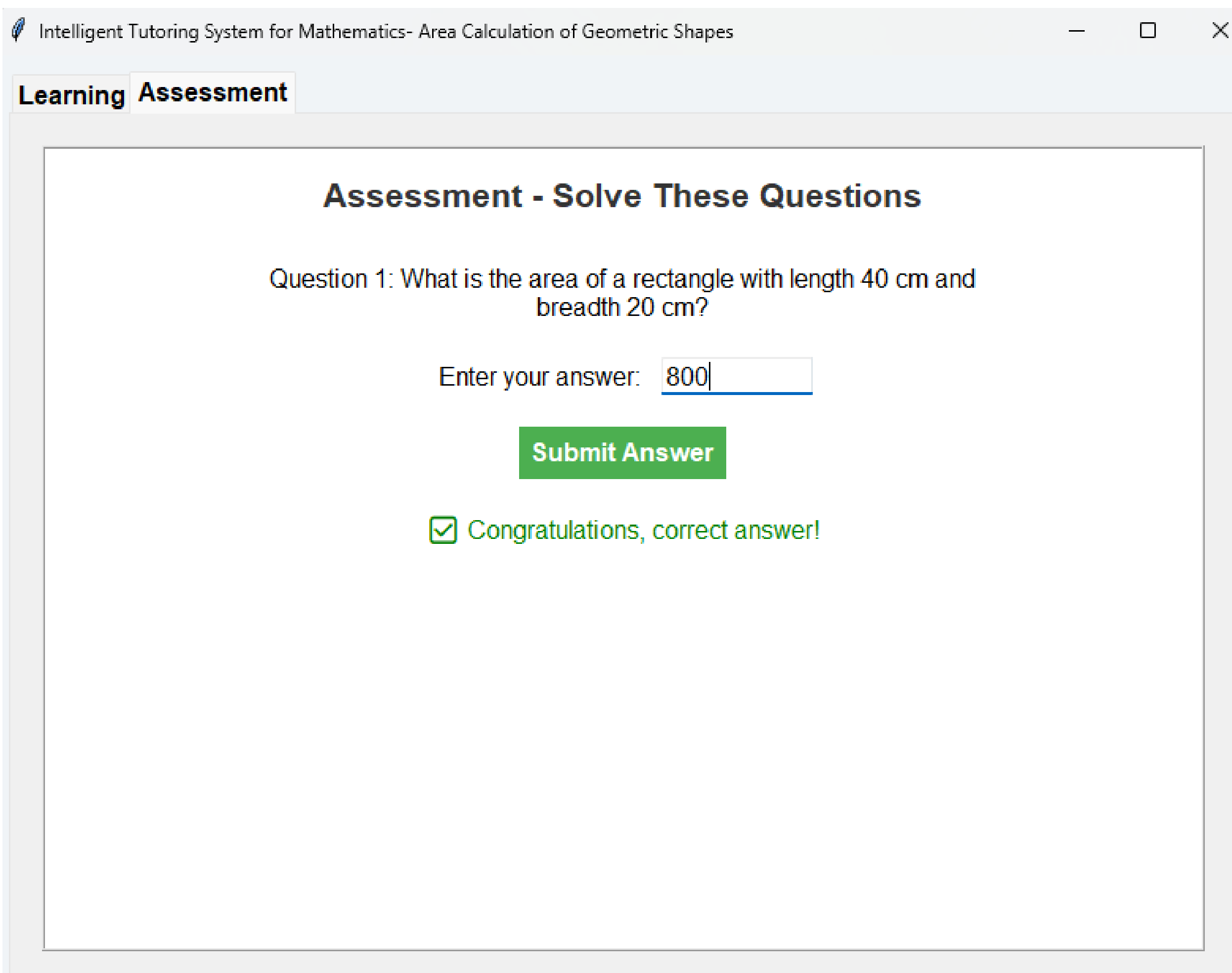

Figure 18: Feedback for Correct Submission

The above figure 18 shows the assessment tab where for the correct submission of answer to the question, the positive feedback is presented in green color.

But if the submitted answer is wrong, we will get feedback as 'Wrong Answer' with the solution of the questions which is shown in the below figure.

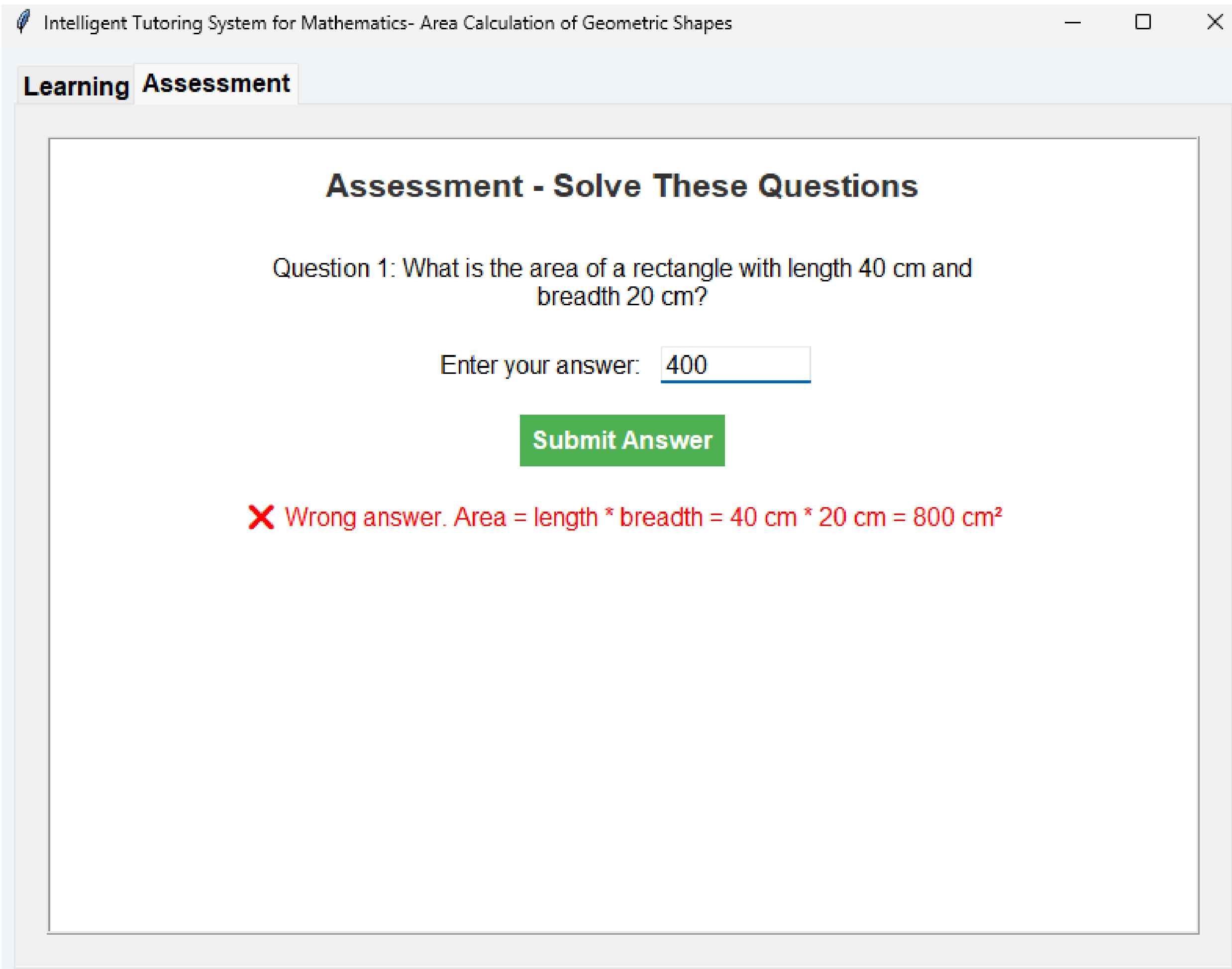

Figure 19: Feedback for Wrong Submission

4. Limitations of the System

While the ITS provides significant educational benefits, it also has certain limitations:

1. **Scope:** The current system is limited to basic geometric shapes (circle, square, rectangle, triangle). More advanced or 3D shapes are not included.
2. **Static Ontology:** Any modifications to the ontology (e.g., adding new shapes) must be done manually in Protégé.
3. **Interface Usability:** While the GUI is functional, it could benefit from improved aesthetics and additional features like animations or gamified elements.
4. **Resource Requirements:** It requires Python runtime environment and libraries like Owlready2 for ontology processing.

CONCLUSION

The development of the Intelligent Tutoring System (ITS) for mathematics, focusing on area calculations of geometric shapes, represents a significant milestone in integrating interactive learning with ontology-based knowledge systems. The project successfully combined a user-friendly Graphical User Interface (GUI) built using Tkinter with an ontology created in Protégé, ensuring that both conceptual and practical learning were addressed.

1. Key Achievements:

- Designed a Learning Module that dynamically displays formulas, accepts user inputs, calculates results, and provides real-time feedback for various geometric shapes such as circles, rectangles, triangles, squares, cubes, and cuboids.
- Integrated a YouTube-based resource system, enabling learners to access additional video tutorials, thus supporting diverse learning styles.
- Developed an Assessment Module with interactive quizzes that assess user understanding, provide detailed feedback, and correct explanations for mistakes, enhancing conceptual clarity.
- Successfully leveraged ontology-based knowledge representation to standardize the structure of mathematical formulas and problem-solving pathways, ensuring scalability and reusability.

2. Challenges and Solutions

- One major challenge was managing the dynamic input fields for varying shapes and their required parameters. This was addressed by implementing a flexible input framework that generates fields based on the selected shape dynamically.
- Ensuring real-time feedback for assessments required careful handling of user inputs and result verification. This challenge was overcome through efficient error handling and informative explanations for incorrect answers.
- Learning how to integrate OWL-based ontology into the ITS was a technical challenge, but it was resolved through extensive research and the utilization of the Owlready2 library, which facilitated ontology loading and querying.

3. Lesson Learned:

Through this project, profound technical and non-technical lessons were learned:

- Technical Skills: The project deepened our understanding of GUI design using Tkinter, ontology development in Protégé, and programmatic integration with Python libraries like Owlready2. We also honed problem-solving skills by handling dynamic input generation and real-time assessments.
- Non-Technical Skills: The project emphasized the importance of user-centric design, ensuring the interface was intuitive and engaging. Effective time management, research, and teamwork were critical in addressing challenges and delivering a robust solution.
- Self-Awareness: This project underscored the need for continuous learning and adaptability, particularly when dealing with complex technologies and frameworks like ontology systems.

4. Future Improvements

While the current ITS meets its primary objectives, there are opportunities for further innovation and refinement:

- **Ontology Expansion:** The ontology can be extended to include additional topics, such as volume calculations, advanced mathematics (e.g., calculus or trigonometry), and real-world problem-solving scenarios.
- **AI Integration:** Incorporating Artificial Intelligence (AI) techniques, such as personalized learning paths or adaptive assessments, would enable the system to cater to individual learner progress.
- **Multimedia Integration:** Expanding multimedia resources, such as animations or interactive diagrams, would enhance user engagement and understanding of geometric concepts.
- **Web-Based Deployment:** Migrating the system to a web-based platform would make it more accessible, enabling learners to use it across devices without requiring software installation.
- **User Performance Analytics:** Implementing a tracking system to analyze user performance, track progress, and provide tailored suggestions for improvement would add value to learners and educators alike.

In conclusion, this project has demonstrated the successful development of an ITS for mathematics that effectively combines technology, user interaction, and knowledge representation. The lessons learned have been transformative, providing insights into both technical development and the broader importance of user-focused design. With future improvements, this ITS has the potential to evolve into a powerful and innovative educational tool, advancing mathematics education and enhancing the learning experience for students worldwide.

REFERENCES

Arroyo, I. et al., 2010. Improving Math Learning through Intelligent Tutoring and Basic Skills Training. *Lecture Notes in Computer Science.*

Binod, K., 2021. *Intelligent Tutoring Systems from Teachers.* [Online]
Available at: https://passionateinmarketing.com/intelligent-tutoring-systems-from-teachers/
[Accessed 17 December 2024].

Chakravarthula, S. R., 2024. *International Day of Mathematics: Nurturing problem-solving skills for success, The Hindustan Times.* [Online]
Available at: https://www.hindustantimes.com/education/features/international-day-of-mathematics-nurturing-problem-solving-skills-for-success-101710326431559.html
[Accessed 18 December 2024].

Chhowdhary, K., 2020. *Rule Based Reasoning, Fundamentals of Artificial Intelligence.* First ed. New Delhi: Springer.

Earley, S., 2024. *The Role of Ontology and Information Architecture in AI, Earley Information Science.* [Online]
Available at: https://www.earley.com/insights/role-ontology-and-information-architecture-ai
[Accessed 17 December 2024].

Huang, X. et al., 2016. Intelligent tutoring systems work as a math gap reducer in 6th grade after-school program. *Learning and Individual Differences,* Volume 47, pp. 258-265.

Laoyan, S., 2024. *What is Agile methodology? (A beginner's guide).* [Online]
Available at: https://asana.com/resources/agile-methodology
[Accessed 18 December 2024].

Liljedahl, P., Santos-Trigo, M., Malaspina, U. & Bruder, R., 2016. *Problem Solving in Mathematics Education, ICME13,* Hhamburg: Springer Nature.

Litman, D., 2016. Natural Language Processing for Enhancing Teaching and Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1).

Niño, F. et al., 2023. Systematic Review: Trends in Intelligent Tutoring Systems in Mathematics Teaching and Learning. *International Journal of Education in Mathematics Science and Technology*, 12(1), pp. 203-229.

Pirvulescu, A., 2020. The Use of Intelligent Tutoring Systems in MathematicsThe Use of Intelligent Tutoring Systems in Mathematics. *Intelligent Tutoring Systems.*

Pressman, R. S., 2014. *Software Engineering: A Practitioner's Approach.* 9th ed. s.l.:McGraw-Hill Education.

Shih, S.-C., Chang, C.-C., Kuo, B.-C. & Huang, Y.-H., 2023. Mathematics intelligent tutoring system for learning multiplication and division of fractions based on diagnostic teaching. *Education and Information Technologies,* 28(2023), pp. 9189-9210.

Siekmann, J. & Erica, M., 2004. ActiveMath: An Intelligent Tutoring System for Mathematics. *Artificial Intelligence and Soft Computing - ICAISC 2004,* Volume 3070.

Simms, V., 2016. Mathematical mindsets: unleashing students' potential through creative math, inspiring messages and innovative teaching. *Research in Mathematics Education,* 18(3), pp. 1-4.

Wang, S., Heffernan, N. & Heffernan, C., 2009. *Student modeling for intelligent tutoring systems, R. Hall (Ed.), Student Modeling: The Key to Individualized Tutoring,* s.l.: Springer.

Woolf, B. P., 2009. *Building Intelligent Interactive Tutors: Student-Centered Strategies for Revolutionizing E-Learning.* s.l.:Morgan Kaufmann.

Woolf, B. P., 2010. A Vision for the Future of Intelligent Tutoring Systems. *International Journal of Artificial Intelligence in Education,* 20(1-2), pp. 35-59.