The Coversheet

Student Name (unless anonymised)	NITESH YADAV
Student Number (as shown on student ID card):	250282436
Word Count / Pages / Duration / Other Limits:	4393 words / 26 pages
Attempt Number:	1
Date of Submission:	12/18/2025

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 your 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):	

Assessor’s Feedback (may be delivered in line with the submission)
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.

Table of Contents

Table of Figures

1. Introduction

Finally, learners usually construct restricted or inflexible representations of mathematics, which limit their flexibility when dealing with situations that differ from standard structures. Children who comprehend mathematics only as "operations equal answer" may fail with activities that demand a more nuanced understanding of equality or number connections. This restricted understanding of addition can impede transfer to related mathematical ideas and more complicated reasoning later in learning, highlighting the need for instructional techniques that extend students' conceptual framework (Fuson, 1992).

To improve reasoning and explanation skills, ITS designers have embraced semantic technologies such as ontologies. Ontology-based ITSs can give more relevant feedback and facilitate deeper conceptual learning since they formalize mathematical ideas, rules, and connections (Anon., n.d.). Ontologies in mathematical addition allow the system to represent numbers, operands, and operations in a systematic manner, allowing for correct validation of learner replies and the creation of explanatory advice. Such techniques show how merging artificial intelligence with knowledge representation may increase the efficacy of mathematics educational systems (AbuEloun, 2017).

The main objective of this project includes:

- Create and develop a mathematical domain ontology to represent fundamental mathematical concepts for addition.
- Implementation of a python based Intelligent Tutoring System with Ontology-based integration.
- Live Feedback and Error-Sensitive Guidance to support conceptual understanding rather than rote memorization.
- Demonstrate how ontology-based systems can help beginners grasp abstract mathematical concepts.

This publication aims to facilitate the practical application of the principles underlying AI-driven learning software and to demonstrate how ontological knowledge frameworks can enhance clarity, consistency, and logical coherence within the educational framework.

2. Project Plan

A carefully structured project plan played a crucial role in ensuring that the development of the Intelligent Tutoring System (ITS) progressed in a systematic and effective manner. Even though the project was undertaken individually, established software engineering practices were consciously applied throughout the planning stage to maintain clarity and control (Dipta, 2025). Particular attention was given to defining a realistic project scope, setting achievable timelines, and minimizing potential risks that could affect development.

The project plan served as a guiding framework by outlining clear action steps, development milestones, and the tools required at each stage of implementation. This structured approach not only supported smooth progress but also helped ensure that the ITS was developed in a disciplined, efficient, and goal-oriented manner.

2.1 Project Methodology

The development of the project followed an iterative methodology inspired by agile principles, which allowed the system to evolve progressively through multiple refinement cycles. While formal Agile practices such as daily stand-up meetings or structured sprint reviews were not practical within a single-developer environment, the underlying philosophy of Agile was still effectively applied. In particular, the emphasis on incremental feature development enabled the system to be built in manageable stages, with each iteration improving upon the previous version (Ajibola, 2024). There was four series of project methodology to develop this project:

Iteration 1 Requirement Analysis and Domain Defining

The given stage was associated with determining the main pedagogical aims of the ITS: empowering the students to compute the addition of two numbers and obtaining feedback in case of misunderstandings. The mathematical terms applicable to the field were established, and that is the addition of two numbers in the context of A and B as the mutual number value/data. Educational needs were also elicited especially the necessity of explicit explanations that reflected the reasoning of a human tutor (Russoa, 2020).

Iteration 2 Ontology design and implementation in protégé

During the second development cycle, attention was directed towards the conceptual design and construction of a domain ontology specifically tailored to mathematical addition, using Protégé as the modelling environment. This stage involved identifying and formally modelling essential mathematical elements, including numerical values, operands, arithmetic operations, equations, and computed outcomes. In addition to defining these components, meaningful relationships between them were established to reflect the logical structure and rules governing addition. This included reviewing class hierarchies, verifying property restrictions, and checking for logical consistency to confirm that the ontology was capable of supporting reliable reasoning and inference within the Intelligent Tutoring System (Kumar, 2013) .

Iteration 3 Developing System Logic using Python and its library.

The third iteration centered on the practical implementation of the system's basic logic using Python and its associated libraries. During this phase, Python was used as the major programming language to transform conceptual design and ontological models into functional system behavior. This stage saw the development and integration of key capabilities such as user input management, response validation, feedback generation, and interaction flow (Lamy, 2017).

Iteration 4 -Integration, Testing and Refinement

During Iteration 4, all previously built modules and components were combined into a coherent system to ensure that interfaces, data transfers, and interactions worked properly. Following integration, extensive testing was carried out, including integration testing to check component interactions, system testing to assess end-to-end behavior, and, where relevant, acceptance testing to establish alignment with user requirements (JSTOR, 2010). Refinements were performed based on test results and stakeholder comments to address problems, enhance functionality, and better align features with project objectives (eduqas, 2024). This iterative process of creating, testing, and refining assured the system's continual improvement and increased overall quality.

2.2 Project Timeline

Phase	Duration
Requirement Analysis	Week 1
Ontology Modeling	Week 2
Python Back-End Coding	Week 3 - 4
GUI Development	Week 5
Integration & Testing	Week 6
Report Writing	Week 7

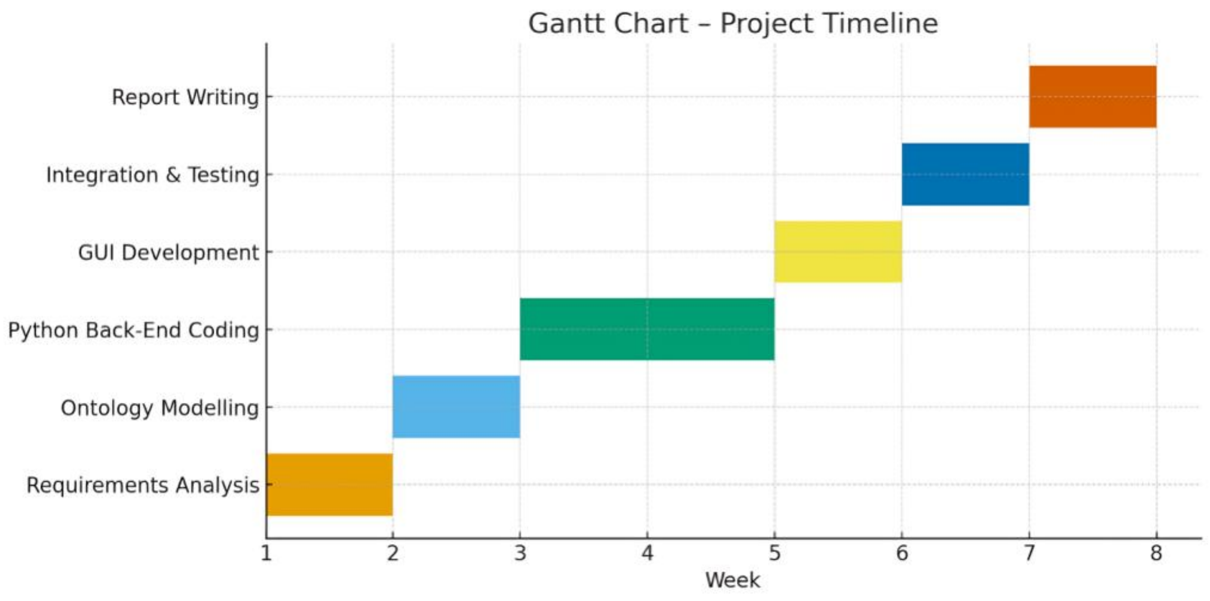
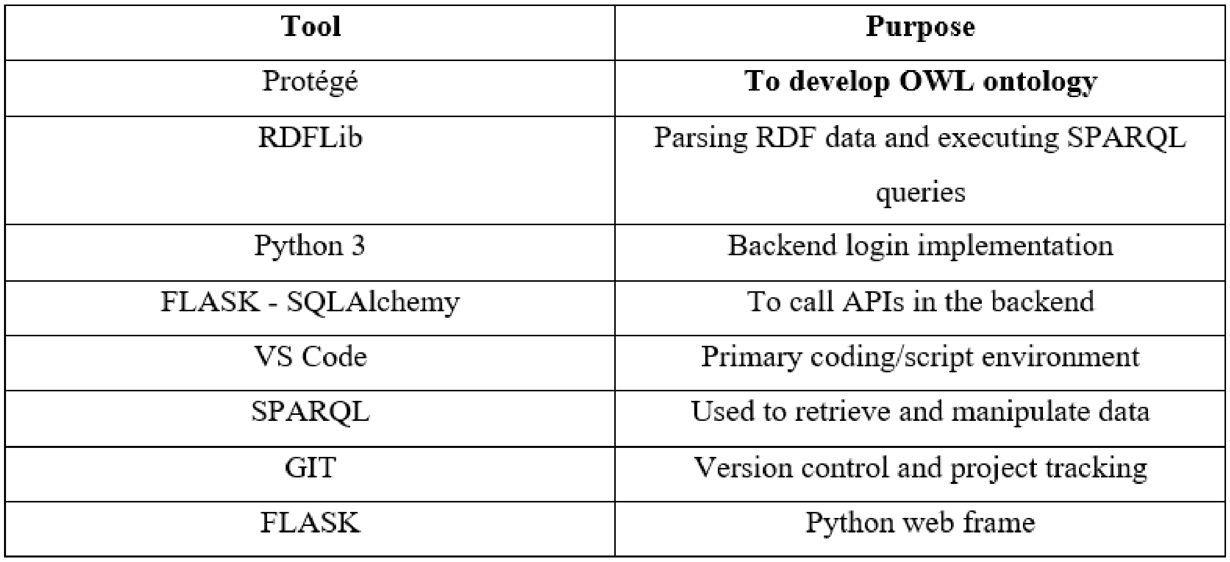


Figure 1 Gantt Chart - Project Timeline

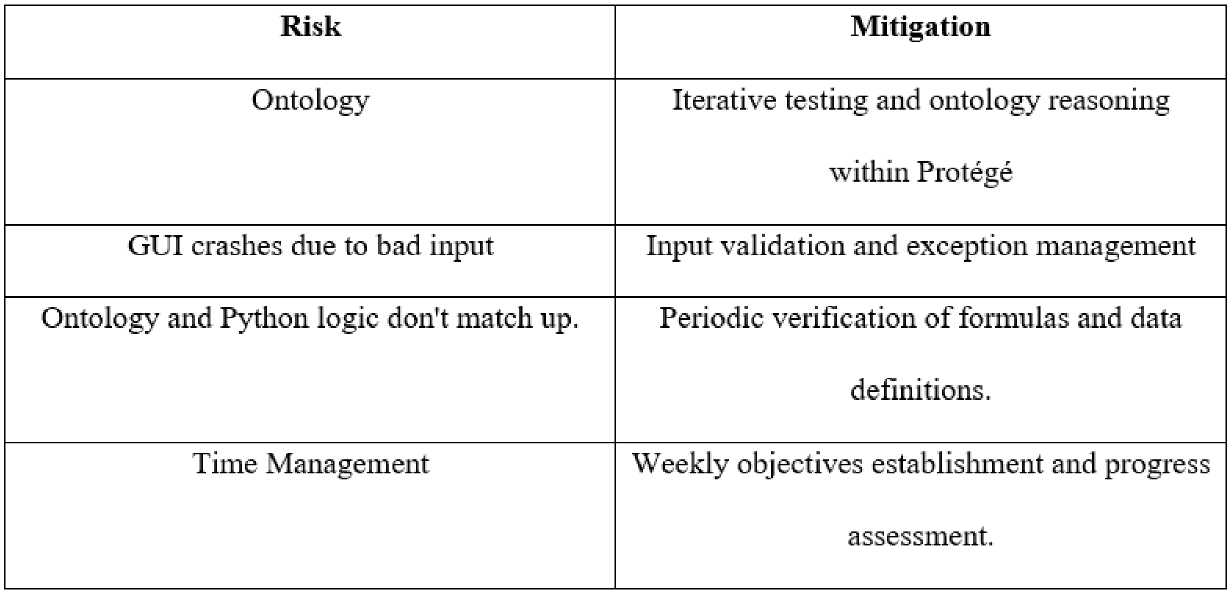
2.3 Tools and Technologies

The mentioned tools and technologies has been used for the development purpose.



Tool	Purpose
Protégé	**To develop OWL ontology**
RDFLib	Parsing RDF data and executing SPARQL queries
Python 3	Backend login implementation
FLASK - SQLAlchemy	To call APIs in the backend
VS Code	Primary coding/script environment
SPARQL	Used to retrieve and manipulate data
GIT	Version control and project tracking
FLASK	Python web frame

2.4 Risk and Mitigation Strategies



Risk	Mitigation
Ontology	Iterative testing and ontology reasoning within Protégé
GUI crashes due to bad input	Input validation and exception management
Ontology and Python logic don't match up.	Periodic verification of formulas and data definitions.
Time Management	Weekly objectives establishment and progress assessment.

3. Literature Review

This literature review addresses the theoretical foundations and technological environment on which Intelligent Tutoring Systems (ITSs) are founded, the use of ontologies in educational software, and the specific pedagogical challenges involved in mathematical addition instruction. The survey explains why a lightweight ontology-based ITS should be created with an emphasis on fundamental mathematical skills.

3.1 Intelligent Tutoring Systems

Intelligent Tutoring Systems (ITS) are advanced educational technologies that employ artificial intelligence to give tailored instruction, closely resembling the assistance that a human tutor would provide to students. These systems assess student replies and give adaptive coaching that allows students to proceed at their own rate, making them especially effective in mathematics instruction where fundamental abilities such as addition serve as the basis for more complicated topics. ITSs may assess student performance in real time and adapt activities to maintain an appropriate level of challenge, increasing learner interest and promoting persistent practice. This individualized adaptability has been well established as a crucial aspect of successful ITSs in mathematics learning settings (Mestre, 2024).

In the context of fundamental arithmetic, such as mathematical addition, ITSs help students by presenting questions, assessing responses, and providing rapid explanatory feedback. This real-time feedback loop helps learners discover faults and comprehend right processes, solving the issue of delayed input in traditional classroom settings. The capacity of ITS to automatically adjust training based on performance data guarantees that each student receives the necessary assistance to develop conceptual comprehension and skill fluency in arithmetic (Chen, 2025).

Although most ITS research has focused on supplementing rather than completely replacing traditional arithmetic education, these systems have demonstrated tremendous promise for transforming learning experiences. Recent studies show that well-designed ITS applications may significantly improve student understanding and autonomy in mathematics learning, particularly at the primary level where core abilities like addition and number sense are important. As ITS technology advances, its adaptive and customized nature is projected to play an increasingly important role in supplementing human instruction and widening access to high-quality mathematics education (Son, 2024).

3.2 Educational Systems using ontology

Ontology-based educational systems provide Intelligent Tutoring Systems (ITS) with a structured and semantic representation of domain knowledge, allowing these computers to comprehend and process mathematical ideas more intelligently. Ontologies formalize the important ideas and connections within a domain, such as numbers, operations, and addition rules, allowing the ITS to reason over this knowledge and facilitate adaptive education (Amin, 2019). According to research, ontology improves educational systems by simplifying semantic querying and conceptual navigation across learning resources, which facilitates personalized learner engagement.

Ontologies improve efficiency in information retrieval and knowledge maintenance in e-learning settings, in addition to providing ITS and feedback. By semantically organizing mathematical ideas, educational systems can more rapidly discover appropriate learning materials or rules for a particular addition issue, simplifying both content delivery and problem validation. This organizational strength also promotes interoperability between learning systems, allowing learning modules to be shared and reused across platforms. Ontology-based e-learning frameworks improve the discovery and reusability of educational information (rajagopal, 2012).

Despite its advantages, developing and integrating ontologies into ITS remains a difficult process due to the requirement for rigorous domain modeling and semantic alignment with instructional methodologies (Chimalakonda, 2020).

3.3 Comparison with Existing ITSs

Existing Intelligent Tutoring Systems (ITSs) vary in design and instructional focus, and not all employ ontology-based reasoning. Classic ITS examples, such as Auto Tutor, engage learners through natural language dialogue and adaptive dialogue strategies rather than formal semantic modeling, emphasizing interactive conversation and cognitive scaffolding across subjects like physics or literacy rather than specific arithmetic domains (Graesser, 2005). These systems employ computational linguistics and pattern matching to emulate human teachers, providing flexible interaction but lacking explicit structured knowledge models for mathematical operations like addition, as opposed to ontology-based ITSs (Aleven, 2016).

Recent ITS research explores hybrid systems that integrate ontologies with AI techniques, such as big language models or individualized student modeling, to obtain both structured domain knowledge and flexible, context-aware coaching. These techniques have the potential to provide more detailed explanations and personalized education by combining semantic reasoning with data analysis (SILVA, 2024).

Ontology-based ITSs for mathematical addition may provide learners with more interpretable reasoning and better conceptual paths than systems relying purely on statistics or procedural feedback. Ontology-enhanced ITSs are a valuable addition to established paradigms, especially for basic mathematics instruction, which requires clear conceptual structure (Silva, 2024).

3.4 Instructional addition calculation: Student challenge

Many students struggle with core conceptual knowledge as well because they focus on memorizing methods rather than internalizing numerical relationships and place value. Without a good mental model of how numbers interact, learners frequently make errors such as misaligning digits or wrongly transferring values between place columns in multi-digit additions (Mangarin, 2024).

Another prevalent challenge stems from misunderstandings about number meaning and grouping, in which pupils may not completely comprehend how digit location impacts value. Research demonstrates that many learners make errors because they do not consistently use place value understanding while adding numbers (Lavidas, 2023). This leads to errors such as adding digits across wrong locations or considering multi-digit numbers as unconnected collections of single digits.

Finally, learners usually construct restricted or inflexible representations of mathematics, which limit their flexibility when dealing with situations that differ from standard structures. Children who comprehend mathematics only as "operations equal answer" may fail with activities that demand a more nuanced understanding of equality or number connections. This restricted understanding of addition can impede transfer to related mathematical ideas and more complicated reasoning later in learning, highlighting the need for instructional techniques that extend students' conceptual framework (McNeil, 2008).

4. ITS Development

This section describes the technical development of the Intelligent Tutoring System, including requirements analysis, ontology engineering, Python integration, GUI implementation, and pedagogical feedback systems. The program was developed in line with core software engineering techniques, ensuring correctness, clarity, and conceptual support across all subsystems (Thevenot, 2023).

4.1 Requirements Analysis

The requirements analysis step identified the ITS functional and non-functional needs. The functional requirements were concerned with the core functions that the system must do in order to promote successful learning. These entailed enabling the user to select a geometric shape, entering the relevant dimensions, and doing a computation. The system needed to retrieve the correct formula from the ontology, rather than using handwritten logic. This aimed to ensure that conceptual knowledge was centralized and accessible.

The Intelligent Tutoring System (ITS) for mathematical addition is intended to show students with dynamically produced addition questions and allow them to enter their responses immediately. The technology evaluates replies in real time and offers prompt feedback, explaining the proper adding procedure when errors occur. It records user efforts and performance, allowing for progress monitoring and step-by-step learning. The interface is meant to be straightforward and user-friendly, allowing learners of all ages to interact effectively with the system. The ITS also facilitates navigation to succeeding questions and provides recommendations when students struggle, so reinforcing conceptual comprehension (Larry, 2016).

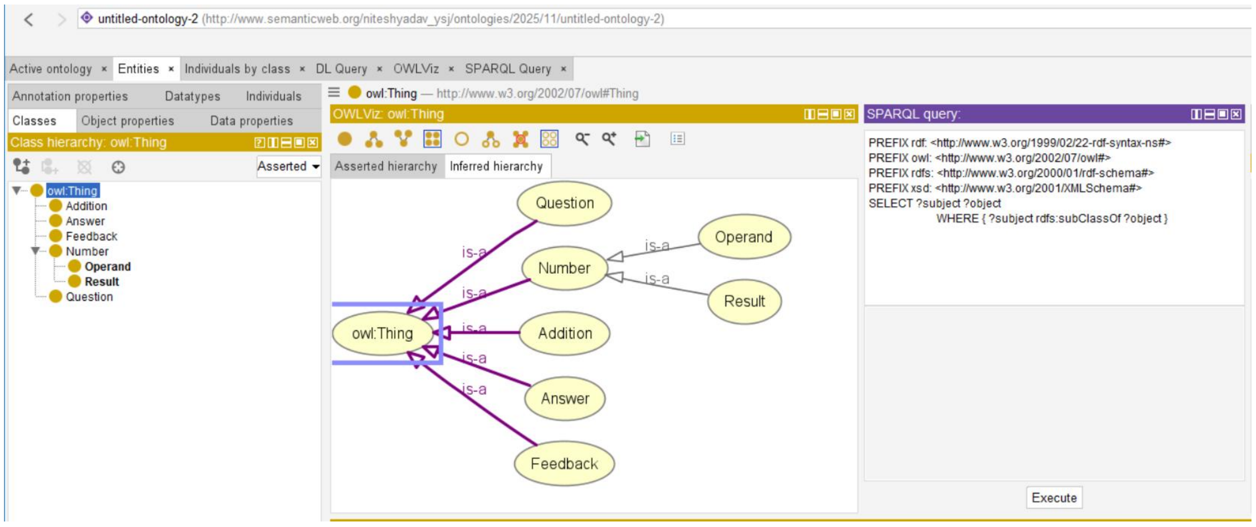
Technically, the system combines Python and RDFLib to provide ontology-based reasoning and reliable validation of addition rules stored in a Protégé-based ontology. Flask is used to develop an interactive web interface that displays questions and accepts user input. SPARQL searches automatically extract appropriate formulae and reasoning rules, allowing the system to deliver intelligent response. Non-functional criteria prioritize real-time responsiveness, robustness against incorrect inputs, scalability for future additions, and general dependability to ensure that students have a consistent and successful educational experience (Achuthan, 2014).

4.2 Ontology Design in Protégé

The ontology for the Intelligent Tutoring System (ITS) in mathematical addition was created using Protégé, a popular ontology building tool that allows semantic knowledge modeling. The ontology's major purpose was to explicitly express the essential ideas involved in simple addition in an organized and machine-interpretable format. Number, Operand, Addition Operation, Result, and Formula are key classes that capture the core parts of addition. Object properties were used to express relationships between these notions, such as connecting operands to an addition operation, whereas data properties were used to hold numerical values and computation results (Archisoup, 2024).

Protégé enables the use of OWL (Web Ontology Language) to construct class hierarchies, restrictions, and connections based on mathematical addition principles. For example, an addition operation was designed to take two or more operands and generate a single result according to preset criteria. Protégé's reasoning assistance contributed to the validation of the ontology by detecting logical flaws and ensuring that relationships were accurately described (L'Heureux, 2009).

The finished ontology was connected with the ITS using Python and RDFLib, allowing SPARQL queries to dynamically extract addition rules and formulae while running. This connection allowed the system to check learner inputs, explain proper answers, and deliver customized feedback using the underlying ontology.

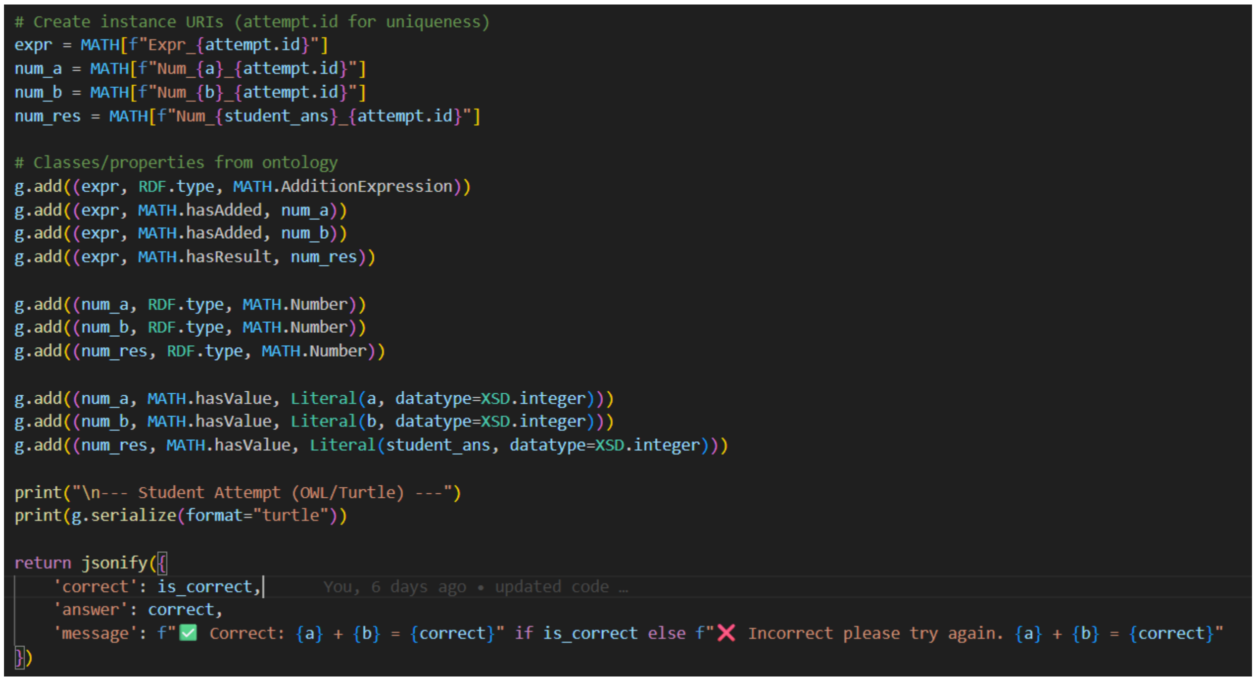


```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?subject ?object
	WHERE { ?subject rdfs:subClassOf ?object }
```

Figure 2 Ontology design in protégé

4.3 Python Integration Using RDFLib

The RDFLib file has been integrated in the python to read and use the methods from ontology.

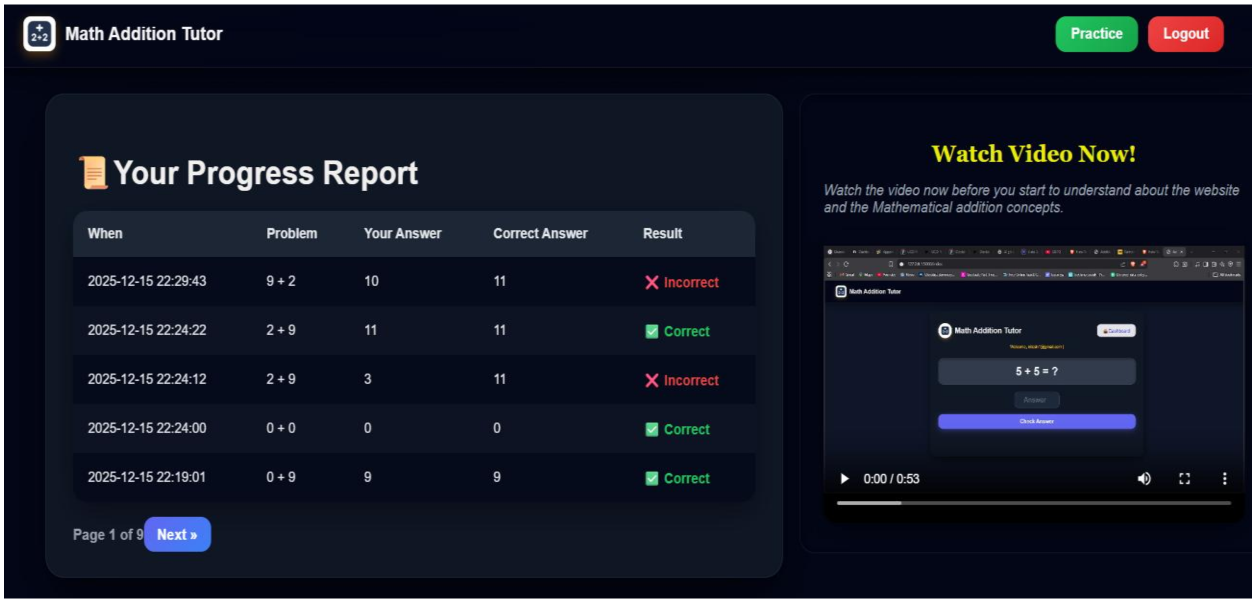


```python
# Create instance URIs (attempt.id for uniqueness)
expr = MATH[f"Expr_{attempt.id}"]
num_a = MATH[f"Num_{a}_{attempt.id}"]
num_b = MATH[f"Num_{b}_{attempt.id}"]
num_res = MATH[f"Num_{student_ans}_{attempt.id}"]

# Classes/properties from ontology
g.add((expr, RDF.type, MATH.AdditionExpression))
g.add((expr, MATH.hasAdded, num_a))
g.add((expr, MATH.hasAdded, num_b))
g.add((expr, MATH.hasResult, num_res))

g.add((num_a, RDF.type, MATH.Number))
g.add((num_b, RDF.type, MATH.Number))
g.add((num_res, RDF.type, MATH.Number))

g.add((num_a, MATH.hasValue, Literal(a, datatype=XSD.integer)))
g.add((num_b, MATH.hasValue, Literal(b, datatype=XSD.integer)))
g.add((num_res, MATH.hasValue, Literal(student_ans, datatype=XSD.integer)))

print("\n--- Student Attempt (OWL/Turtle) ---")
print(g.serialize(format="turtle"))

return jsonify({
    'correct': is_correct,      You, 6 days ago • updated code …
    'answer': correct,
    'message': f"✅ Correct: {a} + {b} = {correct}" if is_correct else f"❌ Incorrect please try again. {a} + {b} = {correct}"
})
```

Figure 3 Python integration using RDFLib file

4.4 GUI development

The graphical user interface (GUI) of an Intelligent Tutoring System (ITS) for mathematical addition is critical to permitting successful interaction between the student and the system. Python with Flask, a lightweight web framework, may be used to create a clean, responsive, and user-friendly interface that is accessible via a web browser. Flask facilitates quick development and smooth connection with backend logic, making it ideal for educational systems requiring real-time interaction and feedback. In an ITS addition system, the GUI is the means by which learners view addition questions, submit their answers, and receive rapid feedback.



Your Progress Report

When	Problem	Your Answer	Correct Answer	Result
2025-12-15 22:29:43	9 + 2	10	11	✗ Incorrect
2025-12-15 22:24:22	2 + 9	11	11	✓ Correct
2025-12-15 22:24:12	2 + 9	3	11	✗ Incorrect
2025-12-15 22:24:00	0 + 0	0	0	✓ Correct
2025-12-15 22:19:01	0 + 9	9	9	✓ Correct

Page 1 of 9 Next »

Watch the video now before you start to understand about the website and the Mathematical addition concepts.

Figure 4 Dashboard GUI

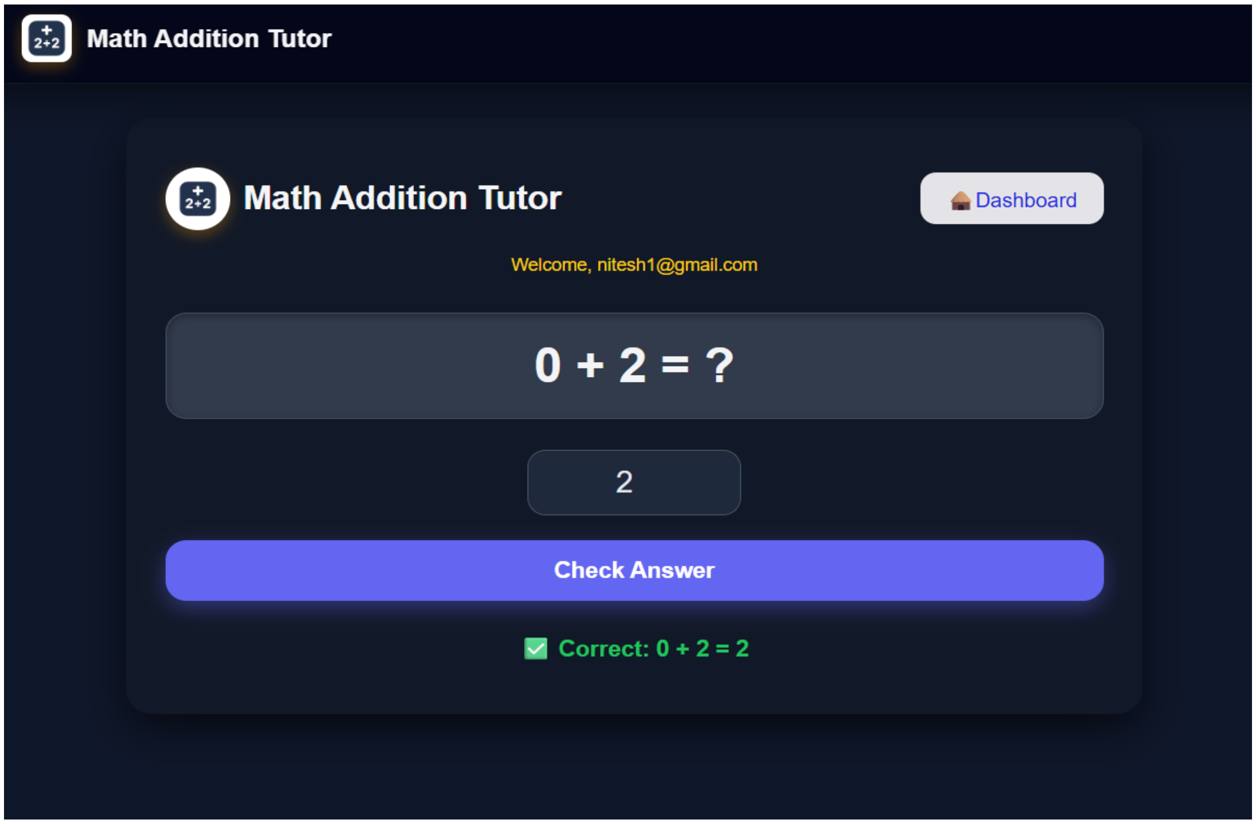
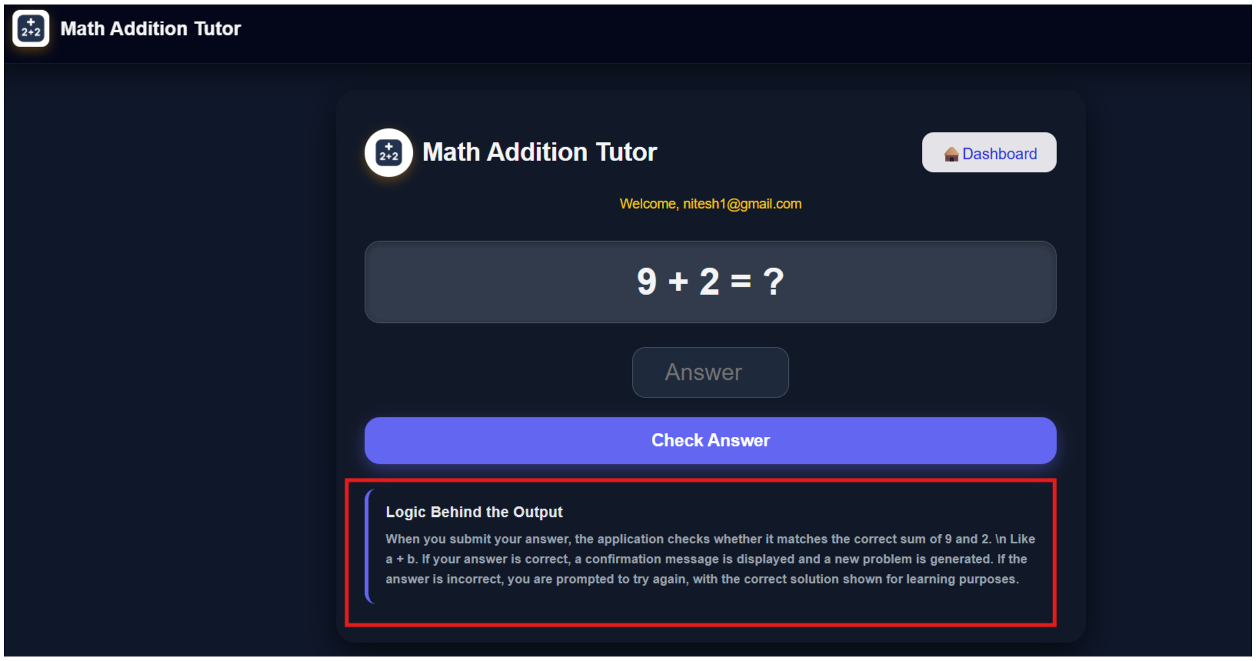


Figure 5 Task GUI

4.5 ITS Feedback Mechanism

The feedback mechanism is an essential component of an Intelligent Tutoring System (ITS) for mathematical addition since it directly effects how learners comprehend and rectify their errors. In an ITS, feedback is provided just after a student provides a response, allowing the system to reply in real time. This immediacy allows learners to notice errors as they happen, reducing the reinforcing of faulty processes. For addition problems, the system compares the learner's input to the right solution recorded in the knowledge base and assesses whether the result is correctly, partially correct, or wrong.



Welcome, nitesh1@gmail.com

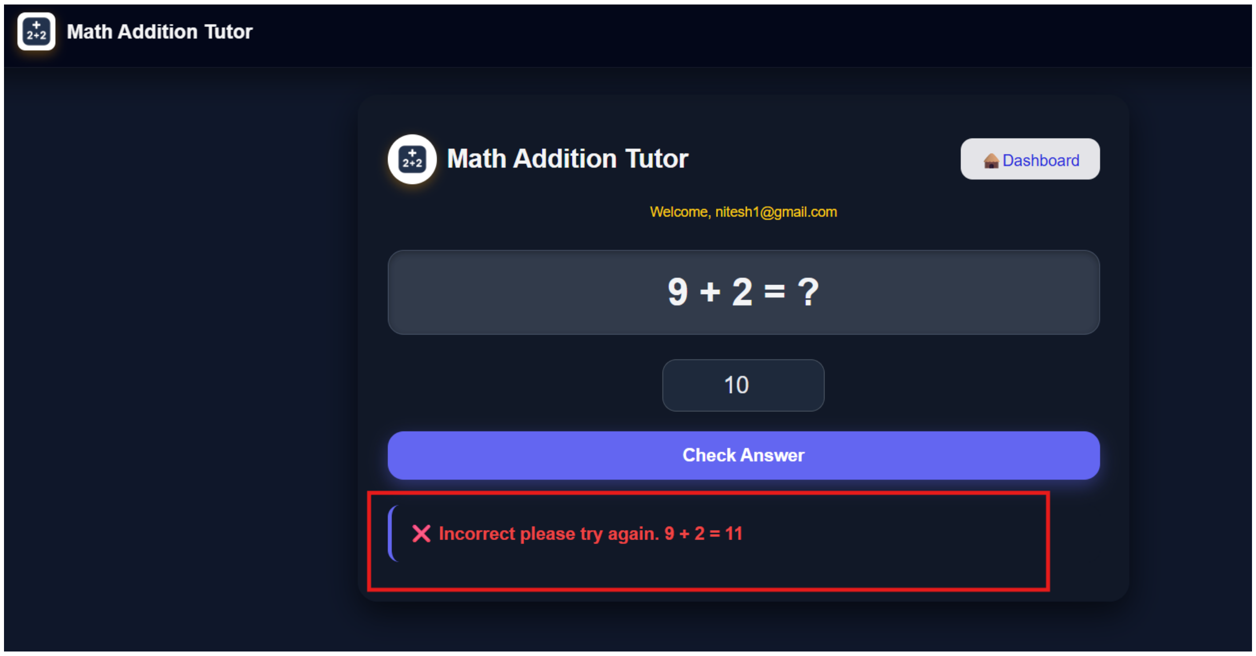
9 + 2 = ?

Logic Behind the Output

When you submit your answer, the application checks whether it matches the correct sum of 9 and 2. \n Like a + b. If your answer is correct, a confirmation message is displayed and a new problem is generated. If the answer is incorrect, you are prompted to try again, with the correct solution shown for learning purposes.

Figure 6 ITS Feedback Mechanism

Error Feedback:



Welcome, nitesh1@gmail.com

9 + 2 = ?

10

✗ Incorrect please try again. 9 + 2 = 11

Figure 7 Error Handling GUI

5 System Evaluation and Reflection

This section evaluates the performance, strengths, and limitations of the produced Intelligent Tutoring System (ITS) and analyzes the development process individually. The review considers the teaching efficacy and technical soundness. This aligns with the module's learning goals.

5.1 System Testing

System testing was performed to assess the Intelligent Tutoring System (ITS) for mathematical addition as a comprehensive and completely integrated solution. At this point, all system components, including the ontology-based knowledge base, SPARQL query processing, reasoning logic, and user interface, were tested in tandem to ensure that they all worked properly. System testing's major goal is to guarantee that the program satisfies its defined criteria and acts as expected in an end-to-end context, rather than focusing just on individual components. This method is widely recognized as a vital stage in the software testing lifecycle (Rossi, 2024).

In addition to functional validation, system testing evaluated usability and dependability. The interface was tested to verify that it remained responsive and simple to use throughout lengthy contact, and feedback messages were reviewed for clarity and instructional value. These tests helped detect minor usability issues and verified that the ITS supported successful learning by providing clear instruction and mistake correction. System testing is vital for validating that educational software provides a reliable, user-friendly, and pedagogically sound experience before deployment or further improvement (Sanz, 2024).

5.2 Strength

The Intelligent Tutoring System (ITS) has several clear strengths that make it especially helpful for learners. To begin with, its simple and user-friendly interface supports students who may lack confidence in mathematics. The clean layout and well-labeled structure reduce confusion, making the system suitable for beginners. In addition, the use of ontology provides a strong and transparent conceptual foundation. By representing formulas and properties explicitly using OWL, the knowledge model becomes easy to review, update, and expand. This approach is more reliable and understandable than hard-coded procedural logic, which is often difficult to interpret and more prone to errors.

Another major advantage of the system is the immediate and high-quality feedback it offers, which is essential for effective learning. Learners are not only shown the correct numerical answers but are also guided through the relevant formulas and conceptual explanations when needed.

Finally, the system's focus on foundational geometry makes it particularly effective for early learners. It helps identify and correct common misconceptions that are often overlooked in traditional classroom settings, thereby strengthening students' understanding from the ground up.

5.3 Limitations

One of the key disadvantages of the Intelligent Tutoring System for Mathematical Addition is its limited topic coverage. The technology is currently limited to simple addition problems and does not enable more sophisticated arithmetic operations or complicated problem solving scenarios. Furthermore, the ontology is confined to core notions such as numbers, operands, and simple addition rules, limiting the system's potential to meet various learner demands or adapt to varying levels of mathematical expertise. The lack of a learner model further restricts customization because all users receive the same feedback regardless of their past knowledge, learning pace, or performance history.

5.4 Reflection

Creating the Intelligent Tutoring System for Mathematical Addition was a valuable learning experience that improved both technical and conceptual knowledge of educational technology. Designing an ontology-based system demonstrated the need of arranging mathematical information in a clear and logical manner to facilitate learning. The addition ontology was created using Protégé and integrated with Python and RDFLib, which increased comprehension of semantic technologies and reasoning mechanisms, notably in terms of producing dynamic feedback and verifying learner inputs. Working with SPARQL queries and ontology integration created hurdles, but overcoming these obstacles improved problem-solving abilities and gave practical insight into how intelligent systems promote learning processes.

The initiative also stressed the importance of usability and feedback in intelligent tutoring systems. Creating an interactive and user-friendly interface showed how real-time coaching, error correction, and step-by-step explanations may boost student engagement and conceptual understanding. Reflecting on the system's outcomes revealed that it accomplished its purpose of assisting learners with basic addition while also suggesting areas for improvement, such as tailored learning through learner modeling and extension to more difficult arithmetic subjects. Overall, the study demonstrated the need of merging AI methods with strong pedagogical ideas to develop effective learning environments.

6 Conclusion

This project successfully showed the design and implementation of an ontology-based Intelligent Tutoring System to help learners comprehend and perform fundamental arithmetic addition. The technology effectively linked conceptual understanding to actual problem solving by combining semantic knowledge representation with an interactive teaching interface. The use of ontology engineering enabled a systematic and logical representation of addition concepts such as integers, operands, operators, and calculation rules, guaranteeing that mathematical information was maintained in a consistent and reusable format. The ontology was constructed using Protégé, which provided a solid framework for capturing addition knowledge in a simple and flexible manner.

The ontology's integration with Python and RDFLib allowed for dynamic querying of mathematical rules as well as reasoning help during tutoring sessions. This enabled the system to extract relevant addition formulae, evaluate learner inputs, and provide meaningful feedback in real time. The user interface was created to be simple and straightforward, allowing students to actively engage with addition problems, experiment with alternative inputs, and instantly see the results of their activities. Such engagement facilitated active learning and improved conceptual clarity.

The system fared well in terms of functional accuracy, convenience of use, and instructional support. The ITS was very good in providing step-by-step instructions, detecting mistakes during computations, and explaining correct addition techniques. Providing feedback and mathematical explanations at the conclusion of each problem assisted learners in reinforcing their comprehension and remembering essential concepts, in line with Intelligent Tutoring Systems' proven principles of timely feedback and conceptual explanation. While the system meets its core goals, further additions might include expanding the ontology to encompass more difficult mathematical operations, including learner modeling to tailor education, and refining the interface for adaptable and accessible learning. Additionally, adopting the system as a web-based or mobile-responsive application would expand its reach and usefulness.

References

AbuEloun, N. N., 2017. *Mathematics Intelligent Tutoring System.* [Online]
Available at: https://philpapers.org/rec/ABUMIT

Achuthan, N. R., 2014. *Non-Functional Requirements Framework: A Mathematical Programming Approach.* [Online]
Available at: https://academic.oup.com/comjnl/article-abstract/58/5/1122/406613

Ajibola, 2024. *Agile Methodology.* [Online]
Available at: https://www.ijsr.net/archive/v13i2/SR24130104148.pdf

Aleven, V., 2016. *Example-tracing Tutors: Intelligent Tutor Development for.* [Online]
Available at: https://files.eric.ed.gov/fulltext/ED618912.pdf

Amin, A. E., 2019. *Building Intelligent Semantic Educational System (ISES) Based on Ontology and Semantic Web Mining.* [Online]
Available at: https://ijicis.journals.ekb.eg/article_62608.html

Anon., n.d. *Intelligent Tutoring System: Learning Math.* [Online]
Available at: https://onlinelibrary.wiley.com/doi/full/10.1155/2021/5590470

Archisoup, 2024. *addition architectural concept.* [Online]
Available at: https://www.archisoup.com/architectural-form

Chen, Y., 2025. *Evaluation of the Application Effect of Intelligent Teaching Systems in Mathematics Education.* [Online]
Available at: https://www.sciencedirect.com/org/science/article/pii/S154810932500004X

Chimalakonda, S., 2020. *An ontology based modeling framework for design of educational technologies.* [Online]
Available at: https://link.springer.com/article/10.1186/s40561-020-00135-6

Dipta, 2025. *A Comprehensive Study of Advancements in Intelligent Tutoring Systems Through Artificial Intelligent Education Platforms.* [Online]
Available at: https://www.igi-global.com/gateway/chapter/363053

eduqas, 2024. *Testing and Refinement.* [Online]
Available at: https://resource.download.wjec.co.uk/vtc/2020-21/KO20-21_1-3/KO20-21_1-3_testing_and_refinement.pdf

Fuson, K. C., 1992. *Research on Learning and Teaching Addition and Subtraction of Whole Numbers.* [Online]
Available at: https://www.taylorfrancis.com/chapters/edit/10.4324/9781315044606-2/research-learning-teaching-addition-subtraction-whole-numbers-karen-fuson

Graesser, 2005. *Running head: TUTORIAL DIALOGUE WITH AUTOTUTOR.* [Online]
Available at: https://bpb-us-w2.wpmucdn.com/blogs.memphis.edu/dist/d/2954/files/2019/10/AutoTutor-chapter-olney_publications.pdf

JSTOR, 2010. *Integration testing.* [Online]
Available at: https://en.wikipedia.org/wiki/Integration_testing

Kumar, J., 2013. *Implementation Of Ontology Matching Using.* [Online]
Available at: https://www.researchgate.net/profile/Priya-m/publication/272709892_Implementation_Of_Ontology_Matching_Using/links/5c55d53692851c22a3a3daeb/Implementation-Of-Ontology-Matching-Using.pdf

Lamy, J.-B., 2017. *Ontology-oriented programming in Python with automatic classification and high level constructs.* [Online]
Available at: https://www.sciencedirect.com/science/article/abs/pii/S0933365717300271

Larry, H., 2016. *Mathematical Underpinnings for Achieving Design Functional Requirements.* [Online]
Available at: https://www.sciencedirect.com/science/article/pii/S221282711630381X

Lavidas, K., 2023. *Misconceptions about Numbers and.* [Online]
Available at: https://files.eric.ed.gov/fulltext/EJ1392724.pdf

L'Heureux, E. G., 2009. *Additions & Subtractions: Studies in Form.* [Online]
Available at: https://issuu.com/ofeqiq/docs/additions_subtractions

Mangarin, R. A., 2024. *Difficulties in Learning Mathematics: A Systematic Review.* [Online]
Available at: https://rsisinternational.org/journals/ijrsi/articles/difficulties-in-learning-mathematics-a-systematic-review/

McNeil, N. M., 2008. *Limitations to Teaching Children.* [Online]
Available at: https://academic.oup.com/chidev/article-abstract/79/5/1524/8274959?redirectedFrom=fulltext&login=false

Mestre, G., 2024. *Trends in Intelligent Tutoring Systems in Mathematics Teaching and Learning.* [Online]
Available at: https://ijemst.com/index.php/ijemst/article/view/665

rajagopal, r., 2012. *Ontology Based E-Learning Systems in the Semantic Web.* [Online]
Available at: https://ijcjournal.org/InternationalJournalOfComputer/article/view/77

Rossi, C., 2024. *A survey of ontology-enabled processes for dependable robot autonomy.* [Online]
Available at: https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2024.1377897/full

Russoa, J., 2020. *An investigation into children's proficiency with simple addition and their flexibility with mental computation strategies.* [Online]
Available at: https://www.tandfonline.com/doi/full/10.1080/10986065.2020.1842968

Sanz, R., 2024. *ontology-enabled processes.* [Online]
Available at: https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2024.1377897/full

SILVA, U., 2024. *ONTOLOGY AND LARGE LANGUAGE MODEL.* [Online]
Available at: https://dl.lib.uom.lk/server/api/core/bitstreams/0fbbe9ce-a1b8-4058-9efe-98a0f9732ccc/content

Silva, U., 2024. *Ontology and large language model based intelligent tutoring system.* [Online]
Available at: https://dl.lib.uom.lk/items/41d1bed7-8bdd-4b78-8147-85b3dd2b3e58

Son, T., 2024. *Intelligent Tutoring Systems in Mathematics Education: A Systematic Literature Review Using the Substitution, Augmentation, Modification, Redefinition Model.* [Online]
Available at: https://www.mdpi.com/2073-431X/13/10/270

Thevenot, C., 2023. *The development of simple addition problem solving.* [Online]
Available at: https://www.sciencedirect.com/science/article/pii/S0022096523000863