1 The Coversheet

Student Name (unless anonymised)	Mohamed Nusly Mohamed Nizam
Student Number (as shown on student ID card):	230273669
Word Count / Pages / Duration / Other Limits:	6550
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 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?*

## 2 **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.	

Table of Contents

List of Figures

1. Introduction

In today's rapidly evolving technological landscape, the education sector has been profoundly influenced by advancements in artificial intelligence (AI). Of these innovations, ITS has turned out to be effective tools for delivering personal, flexible, and self-paced training. In contrast to conventional classroom practices where common practices are considered appropriate for all learners, ITS was designed to meet the specific learning requirements, past knowledge, and other learning modalities of each learner, thus overcoming the shortcomings of regular academic programs. For learners to attain more and become more responsive in various domains learning has to becomes more responsive.

In the past, various ITS have been developed in various fields to simulate a human tutor. These systems allow students to 'study at their own pace' and get help and guidance, clear instructions and immediate feedback from the systems based on use of AI. This paper discusses the development of an ITS that would help teach basic geometry in math, specifically, how to find the areas of triangles, rectangles, and circles.

The use of AI methods is important to address the problems that limit the efficiency of learning geometry and help to personalize the access to knowledge. For example in Adaptive Learning Algorithms, the lessons and problems are mastered and adapted to give each of the students the pace at which they are supposed to learn (Raghu Raman; Prema Nedungadi, 2010). It isolates students to the right level of difficulty that will enable them cover some grounds without feeling like they are in over their heads. Another approach to AI is that AI can use the means like NLP to analyze the questions of the students and give relevant and satisfactory explanations. Moreover, such machine learning is used to study the students' performance and find out potential weak spots in order to provide specific recommendations how to deal with them. Compositively, these AI tools improve the learning-teaching activities in the context of geometric learning.

In general, Math, particularly geometry, can be problematic for learners because it entails the application of numerous formulas that learners rarely comprehend. To make learning easier, this ITS uses two key ideas: On the other hand, there is an ontology-based knowledge representation and interactive design proposed in this article. These approaches involve the simplification of the content into small and easy to understand lessons. The ontology of the system developed using the Protégé tool and Web Ontology Language (OWL) presents the relations between the main topics such as shapes, formulas, and units and how they are dependent on each other. The interface is created in Figma, so students will not have difficulties with following the instructions and engaging in the learning process.

Through this project, we hope to demonstrate how ontologies and AI-ITS, combined with search algorithms like BFS, can complement the traditional models of learning to make learning easier, efficient and learner-centered.

2. Project Plan

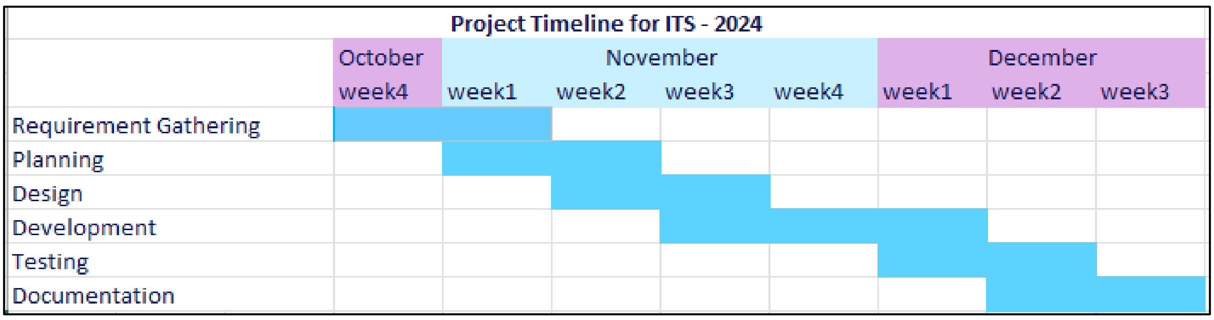
2.1 Objectives

- Create an ITS that both educates and facilitates students to solve simple geometric shapes' area.
- Create a domain-specific ontology about geometry concepts by using Protégé tool.
- Develop an easy to use interface in Figma to engage with the ontology and give instructions on how to learn.

2.2 Roles and Responsibilities

- System Architect: Explain the architecture of the ITS and be accountable for the components of the ITS.
- Ontology Developer: The domain ontology is developed in Protégé and contains shapes, formulas, and examples.
- Interface Developer: Create a simple user interface using Figma.
- AI Specialist: Ensure that the system uses ontology-based feedback to build logical arguments that will guide the students.
- Tester: This means that the ITS must be usable, accurate and robust.

2.3 Milestones



- October (week4) and November (week1) : Research and finalize the requirements.
- November (week1 and week2) : planning the ontology domain and sketch the prototype design.
- November (week2 and week3) : Design a High Fidelity User Interface (UI) prototype in Figma and Functional Interface with python.

- November (week3, week4) and December (week1) : Develop the ontology in protégé and Integrate the UI with the ontology using python.
- December (week1 and week2) : Testing
- December (week2 and week3) : Complete the documentation report.

The mentioned works are divided into steps to reflect the progress of the project. The first of them is the phase that takes place at the end of October and the beginning of November, which is devoted to the collection of requirements and planning, which forms the basis for further work. Mid November is the design phase where the structure of the system and the user interface is created. In the course of this, by late November, the development phase is set to start; The educational materials are incorporated, as with such features like security. The testing begins in early December and is a form of check point to review the early versions and get feedback. Last is documentation which completes the project in mid December whereby all associated activities are documented in preparation for the final delivery and deployment.

2.4 Project Management Approach

Organize work in a week according to the Agile methodology, directing tasks into sprints. This makes the work progressive, in the sense that feedback can be easily incorporated as new rounds are worked through.

The Intelligent Tutoring System will utilize an Agile methodology to cultivate this proposed framework. Separating execution into sprints through an iterative and progressive strategy will permit productive advancement. This strategy diverges from conventional plans as testing happens all through, not at just completion. Agile permits adaptable alteration as circumstances change, opening entries to progress. There is no single answer for structure an application. Diverse serendipitous and enlightening ways may create and convey results over web or portable stages. Some hypothesized outlines could streamline organized instruction, yet nonspecific unpredictable sentences likewise engage experimentation.

The advantages of using the Agile methodology when developing an Intelligent Tutoring System include maintaining flexibility throughout the iterative development process, the ability to collaborate closely with potential clients as requirements evolve, decreased emphasis on extensive documentation, and incorporating late-arriving requirements.

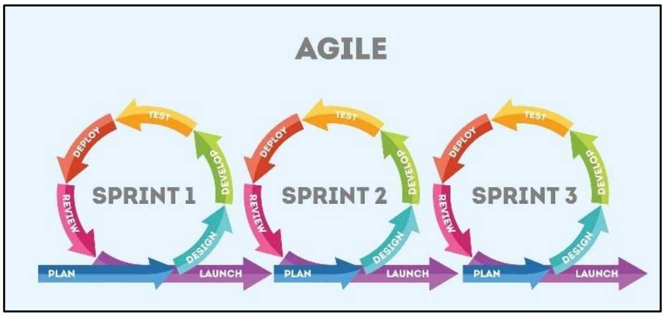


Figure 1: Agile Methodology (Slawek-Polczynska, 2020)

As depicted in Figure 1, sprints are central events in the Scrum framework. The initial design concept is intended to evolve continuously as the project progresses through multiple development cycles. Within each sprint, teams modularize the scope of work and focus intensely but briefly on planning, analysis, design, coding, testing, and integration of discrete features before gathering feedback to guide future enhancements. Once all sprints conclude, the system is considered complete and ready for launch, though refinement may continue in response to end user input identifying areas for improvement. Thus, through recurring, rapid sprints wielding collaborative adaptation to emerging needs, Agile cultivation nurtures intelligent tutoring solutions that dynamically align with a shifting learning landscape (Agrawal, 2022).

3 Literature Review

The Intelligent Tutoring System (ITS) is a computer programme that is designed to educate students through providing them with feedback, instructions and tests without the help of a tutor. Over the last few years, much has been done to develop ITS that would mimic the behavior of a human tutor. Some of the drawbacks of ITSs that have been developed include the fact that they cannot be used to share courses across different domains and the fact that it is very difficult to develop ITSs. There is a line of work that promotes ontologies as the systems for representing domain knowledge.

ITS are usually domain specific, that is, they are designed to teach specific subject areas. However, they should be used by the different category of learners with different levels of prior knowledge, abilities or psychological profile. To enhance the reusability of created knowledge models and therefore minimize the development costs, it is necessary to achieve the maximum degree of separation of learning domain dependent and learning domain independent reasoning and tutoring methods.

According to (Peter Brusilovsky, Elmar Schwarz, and Gerhard Weber, 2023) an Intelligent Tutoring System (ITS) should offer lessons, instructions, questions, and feedback for every learner while evaluating the learner’s behaviors so that it can be interpreted using different graphical and textual means. It also explains how the past applications of ITS cannot be reused as they are mostly isolated depending on the context. According to the authors, ontologies can be used to build relationships among different concepts, although it is hard to maintain and enhance. The paper suggests the domain dependent components to be separated as much as possible from the domain independent components to enable more repetitions in the ontologies regarding Intelligent Tutoring Systems (ITS).

A student model, an educational module with all the information in relation to various disciplines, a mode that facilitates user-system connection, and a domain module with all the knowledge (educational content) comprise traditional ITS. An ontology for students using the ITS platform is proposed by (Panagiotopoulos et al., 1970). Students' academic and personal information are the two categories of student data that the ontology uses. Per the authors, the suggested ontology, irrespective of its domain, would prove useful in future ITS. Accessible through a web application, the created student profile ontology can serve as a crucial information technology system module. Moreover, the writers intend to include additional guidelines.

Creating very compatible Semantic Web ontologies that can be linked to other The subject domain and the student model that are part of the ITS are among the best ways to officially prescribe educational norms. (M. Chang, G. D'Aniello, M. Gaeta, F. Orciuoli, D. Sampson and C. Simonelli, 2020) employed data mining techniques for the purpose of automatically generating and refining rules from real-life tutoring interactions as well as describing how the use of the Semantic Web-based approach is advantageous in defining instructional rules for (M. Chang, G. D'Aniello, M. Gaeta, F. Orciuoli, D. Sampson and C. Simonelli, 2020) suggest a new approach that eliminates its main disadvantages of conveying it through creation of web based expressions and ultimately presenting it using Web Ontology Language (OWL). An ontology language One of the advantages of the tutoring model is that the rules of the model can be represented using (OWL). suggested technique. In fact, the CPAR data mining method can be used to extract ontology rules. provided. The extraction of rules from data collected from actual tutoring interactions is made possible by this technique. To overcome the problem of uneven rules and default actions, the authors plan to implement more complex solutions in their future work.

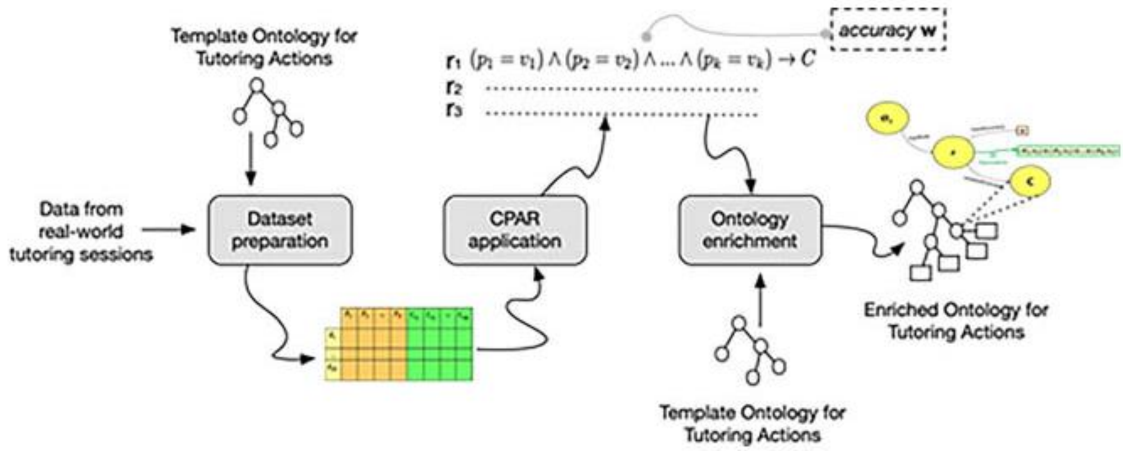


Figure 2: The Ontology Enrichment approach based on Predictive Association Rules (CPAR) to build a tutoring system by using data from real-world tutoring sessions (M. Chang, G. D'Aniello, M. Gaeta, F. Orciuoli, D. Sampson and C. Simonelli, 2020).

(Peter Brusilovsky, Elmar Schwarz, and Gerhard Weber, 2023) proposed a LISP teaching approach in ELM-ART for learners who get more support from virtual teaching assistants. In addition, the system would automatically change the level of difficulty depending on the learner. capacity. However, since there was no learning assistance available, this system in essence, it provided feedback that was as elementary as what the teachers provided.

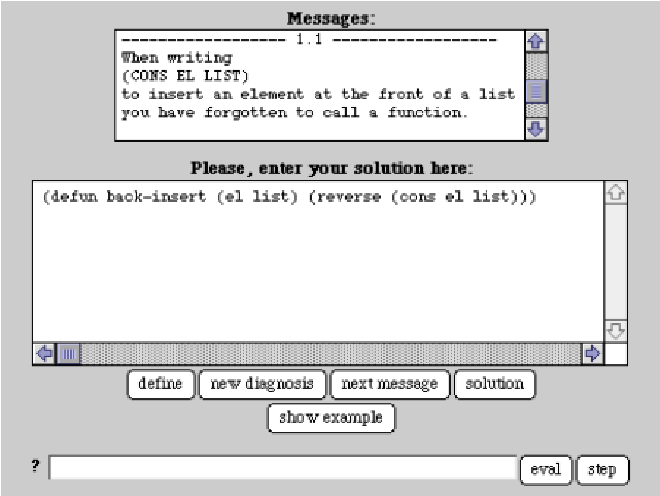


Figure 3: Form-based interface for problem solving support (Peter Brusilovsky; Elmar W. Schwarz;Gerhard Weber, 1996)

In different fields of study, the term data model (DM) is used to provide a conceptual model of the behaviour, structure or other characteristics of a particular entity. DM as a conceptual model is responsible for the organization of knowledge structure, relations, semantics and. consistency requirements. According to (Stankov, S., Glavinic, V. and Rosic, M., 2010), a domain model refers to a set of knowledge regarding a particular topic, concept or discipline. Artificial intelligence appears as a technology that can collect and analyze information. within this context. In this case, the Data Knowledge Model (DKM) is the model that is used. knowledge organization system within a given subject area, especially used as a part of online tutoring programs. The proposed system offers the user a chance to access many Multiple-Choice Question (MCQ) assessments. After that, the system provides feedback to students depending on the answers provided, explaining the extent of the accuracy of the answers. Updates and recommendations are then adjusted according to the student's performance. These insights include recommended reading. works, questions to tasks, correct and incorrect answers given by students, their total performance, and the changes in their knowledge level. To proceed to the next task question, students are to choose the right answer, which corresponds to the topic of their thematic interest. instruction. The system assesses the correctness of the response and the response is followed by the feedback to the learner based on the performance of the learner in the course.

For online instruction in computer programming languages, the Expert Tutoring System (E-TCL) was developed represents a collaborative effort among numerous educators (M.M El-Khouly, B.H Far and Z Koono, 2000) . This collaborative initiative was to ensure that one or more of the goals of the health care system were met. integrate additional computer programming languages even more smoothly into the curriculum, encouraging efforts that ranged from altering and expanding taught commands to inventing multiple tutoring dialogues for a single command and the use of different tutoring. techniques. Similarly, students used the WWW to interact with the system, the freedom to choose the programming language of their preference, deciding the best way to present information and engaging in the sharing of learning experiences. with peers. This innovative system introduced three distinct types of agents: Tutoring Agents Teacher's Assistants (TA), Personal Assistant Agents for Teachers (PAA-T), and Personal Assistant Agents for Students (PAA-S), all of which have a user interface that can be personalized. Facilitated by HTTP and IIOP protocols, the client-side PAA-T and PAA-S successfully connected to the server-side TA. This architectural design not only enabled teachers and students to adapt PAA-T and PAA-S functionalities to their needs while at the same time. has been able to effectively manage the demand for servers. Notably, this architecture allowed for having as many teacher agents to effectively respond to one or more student agents, as a demonstration of adaptability. and effective and efficient teaching and learning model in the area of computer programming language.

Cognitive Tutors in geometry assist the students in learning through the use of a personal tutor. They take learners through problems solving processes and provide clues and feedback where necessary. These systems are based on the idea of how students approach problems, how they err, and how they can be helped to correct their errors. In geometry, they help students learn shapes, formulas, and calculations in areas of triangles or circles amongst others, by alerting each learner’s rate and extent of understanding thus making the lesson delivery efficient and involvement (Corbett, 2001).

The next literature review focused on the GeoGebra. It is a another interactive Software tool. In that, anyone can learn geometry with ease. It enables the student to work through shapes constructions and theorems in a flexible appear and disappear manner. The users have the ability to sketch geometric shapes such as triangle, circle, polygons and the relationships between geometrical properties such as angles and area are dynamically shown as the users change them. It appears that such an approach is beneficial to the students in an effort to grasp the ideas and connection preached in geometry. GeoGebra also helps educators by providing features that teachers could use to design a powerful lesson and simulations (Velichova, 2011).

The challenges that learners in the Pacific region faced when studying computers programming were discussed by Sharma and Harkishan as follows: To effectively address these problems, the present literature review highlighted the deficiencies, which are inherent to the conventional paradigms of teaching and learning and underlined the importance of developing innovative approaches in the sphere, such as intelligent tutoring. systems (Sharma, P. and Harkishan, M., 2022). Also, there is real-time performance and feedback mechanisms along with timely feedbacks. and individualized learning were some of the key elements of effective intelligent tutoring systems that Sharma and Harkishan discussed above. The review of the literature synthesised literature on how these elements could enhance learning outcomes in the field of Computer programming learning.

In order to keep improving an adaptive web-based intelligent tutoring system, the authors of the research (Kularbphettong, K., Kedsiribut, P. and Roonrakwit, P., 2015) reported first findings demonstrating significant development in a prototype model. According to the study, using this approach in various courses could be effective and give students the chance to develop their programming skills. Furthermore, the system's ability to assist instructors in supervising and handling programming classes is well-known. The authors indicate that they are interested in investigating several approaches to improve this project in their upcoming studies.

With the purpose of teaching Android application development, the research by Rekhawi offered a thorough investigation of the theory and architecture of an Intelligent Tutoring System (ITS). The aim of the ITS (Rekhawi, H.A.A. and Naser, S.S.A. , 2018) was to improve the skills and expertise of instructors and students in this field while addressing frequent obstacles encountered in the educational process. Training resources designed especially for Android application development were integrated within the system. Good feedback from participants suggested that testing the planned ITS with instructors and students produced positive outcomes. To build practical and efficient solutions, the conclusion emphasized the significance of continuing research in the fi1eld of artificial intelligence and Android application development. Moreover, it highlighted the benefits of using intelligent tutoring systems widely in the instruction of modern programming languages.

The use of Artificial Intelligence, Cognitive Science, and the educational paradigms found in ITS has received interest in the current society, providing new approaches to quality learning. ALEKS is an intelligent tutoring system used to teach math questions, for which the system is based on the knowledge space theory, illustrating the effectiveness of IT learning environments (Evans, 2024). However, unlike systems intended to support programming languages, such as C# (Al-Bastami & Naser, 1970), which are aimed at improving students' performance via open and forgiving environments, ALEKS has limited guidance related to topics such as geometry. Although the architecture of ALEKS is designed to individualize mathematics instruction, the integration of graphics could enhance the learning process for students, consistent with the literature on ITS to improve learning (Evans, 2024).

This study by Yılmaz examines the effects of integrating ChatGPT into university programming instruction on students' self-efficacy in programming, computational thinking, and course motivation (Yılmaz, F.G.K. and Yılmaz, R., 2023). Because of a control group that participated in a pretest-post-test in the experimental design, the experimental cohort used ChatGPT during programming classes whereas the control group did not. According to the findings, computational thinking skills, programming self-efficacy, and general motivation all increased statistically significantly among students utilising ChatGPT. Sub-dimensions of creativity, algorithmic thinking, cooperativity, critical thinking, and problem-solving abilities on the assessment scales showed significant variations. Between the two groups, there was no statistically significant difference in the sub-dimension of difficult goals. This shows that the use of AI tools may not have a major impact on students' motivation to tackle challenging programming difficulties.

The study focuses on how the use of AI tools such as ChatGPT can be integrated into It is therefore important to understand how programming education can improve the learning process. It underlines the fact that in order to to make the best use of the tools, pupils must have proficient timely writing skills. Teachers should ensure that their pupils are able to write prompt and also ensure that they are AI literate. Metacognitive prompts are one of the most important strategies for encouraging critical thinking, and teachers are suggested that these strategies should be employed to manage student learning as intended. The study's recommendation for enhancing the benefits of implementing ChatGPT and other similar. the integration of technologies into programming education is best done through the use of a holistic approach. f STEM literacy and quick writing skills, four metacognitive and five AI literacy subthemes emerged.

Thus, the literature review provides an extensive description of the state of the field. on ITS research, which stands for Intelligent Tutoring Systems. Some of the topics that were covered in the study are as follows student modelling techniques, feedback system effectiveness, and learning adaptation techniques. It defines shifting dynamics on individualist learning and the impact of technology. field of students' performance and motivation. Furthermore, the combination of several studies stresses the further efforts to enhance the efficiency and usability of ITS stands for Intelligent Transportation Systems. The conclusions made in the review contribute to the development of the field while leaving the door open for additional studies that would concentrate on enhancing the educational outcomes of intelligent tutoring systems.

4 System Analysis

The goal of the project being proposed is to design an intelligent tutor that will help teach students geometry basics. This system is designed to transmit knowledge in areas of concern like the area of triangles, rectangles, and circles. The final categories implemented are intended to allow users to quickly review the fundamental concepts of these objects, and then move on to master more complex topics if desired, such as volumes of pyramids and cones or trigonometry.
The educational content for the system includes 3 levels, each of which is accompanied by a quiz. In this report, the content and structure of each level will be described.

4.1 Modelling in Protégé

The ontology for this system was developed using Protégé, where classes, sub-classes, data properties, and object properties were defined to reflect the structure of geometric learning.

4.1.1 Class Hierarchy

The main structure of the geometry ontology consists of Three primary components, which serve as the foundational classes:

- Geometry: Represent the Geometry Subject of contents
- Student: Focuses on adapting to unique learning objectives and skills of individuals to ensure optimal learning outcomes.
- Pedagogy(Learning Approaches): Describes strategies to deliver knowledge effectively. Includes levels, quizzes, and feedback to evaluate and strengthen understanding.

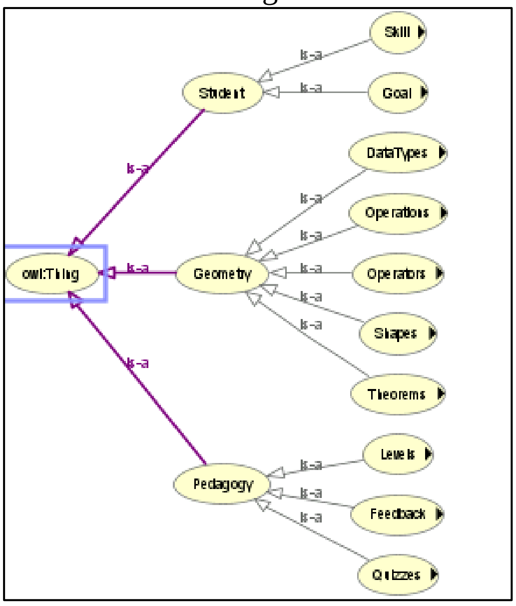


Figure 4: Class Hierarchy

More over, Geometry divided subclasses in to

- Shapes: Represents various geometric entities.
- Theorems: Captures the principles and rules of geometry.
- Operations: Represents mathematical computations and transformations applied to geometric entities.
- Operators
- Datatypes

4.1.2 Sub-class 'Shapes'

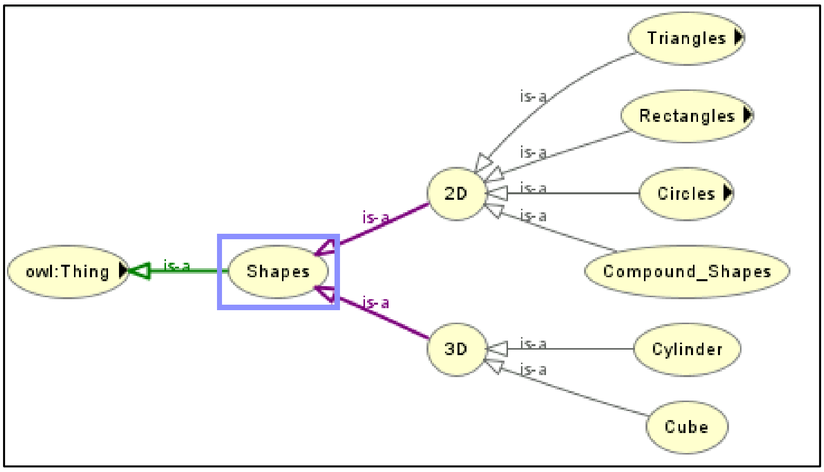


Figure 5: Shapes class

Sub-class 'Shapes' The Shapes subclass is further divided into two major groups in geometry.

1. 2D Shapes
2. 3D Shapes

2D Shapes is further divided into 4 subclass, they are

- Triangles – Recalling the forms of the triangles and, calculating their areas with reference to the base and height.
- Rectangles – Concerned with the kind of dimensions and extending; computation of area through length and width.
- Circles – Introduction of radius, diameter, and the computation of the area choosing the area formula randomly πr2 .
- Compound Shapes – A lesson on how to find the area of compound shapes by using the areas of simple shapes.

3D shapes is further divided into 2 subclass. Which are

- Cube
- Cylinder

4.1.3 Triangles

The Triangles topic involves giving students understanding of basic summit which are necessary in calculating areas of triangles.

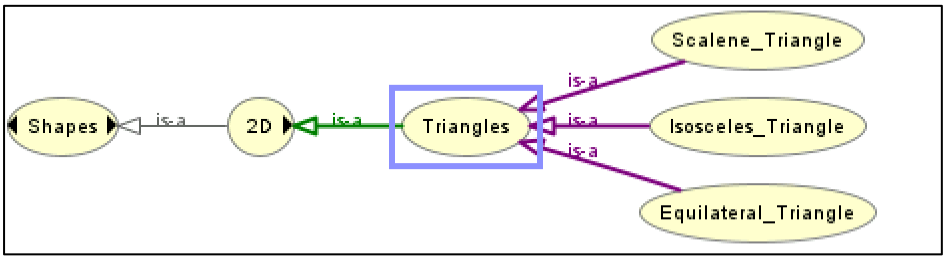


Figure 6: Sub-Class of Triangles

Subclasses include:

- Equilateral Triangle: All sides are equal, and the area is calculated using the formula
  $$\boxed{\text{Area} = \frac{\sqrt{3}}{4} \times \text{side}^2}$$
  Instances: < EquilateralTriangle1>

- Isosceles Triangle: Two sides are equal, and the area is calculated using a combination of base and height.

  Instance: <IsoscelesTriangle1>

- Scalene Triangle: All sides differ, and the area is computed using the formula
  $$\boxed{\text{Area} = \frac{1}{2} \times \text{base} \times \text{height.}}$$
  Instance: < ScaleneTriangle1>

4.1.4 Rectangles

The Rectangles topic shows how to calculate the area of rectangles using length and width.

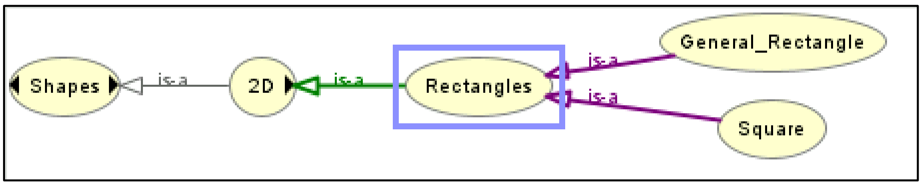


Figure 7: Sub-Class of Rectangles

Subclasses include:

- Square (Special type of rectangle): A special type of rectangle where all sides are equal. The area is computed using.
  $$\boxed{\text{Area} = \text{side} \times \text{side.}}$$
  Instances: < Square1>

- General Rectangle: Represents rectangles with differing dimensions, and the area is calculated using.

$$\boxed{\text{Area} = \text{Length} \times \text{Width.}}$$

Instance: < GeneralRectangle1>

4.1.5 Circles

The Circles topic is centered on radius, diameter, and area.

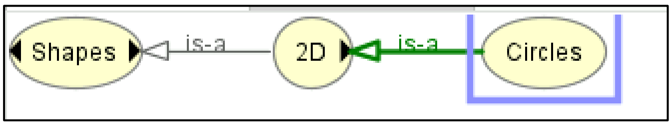


Figure 8: Class of Circles

Attributes include:

- Radius – Creating the instantiation of the key property of the circle.
- Diameter – Double the radius (Diameter=2×Radius).
- Area Calculation – Using the formula πr2.

Instance: <Circle1>

4.1.6 Theorems

This class defines the basic concepts and guidelines used in geometry.

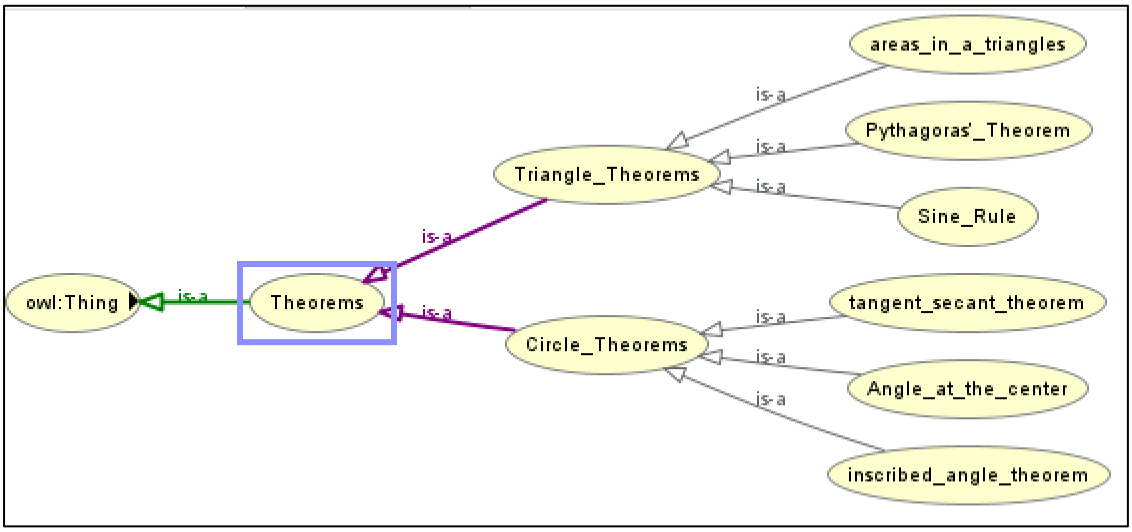


Figure 9: Sub-Classes of Theorems

Sub-classes include:

- **Triangle Theorems**: Formulas like Pythagoras' Theorem, Sine Rule, formulas for areas in a triangles and so on…
  Instances:< PythagorasTheorem>, < SineRule1 >

- **Circle Theorems**: These are; Angle at the center, The tangent secant theorem and the inscribed angle theorem.

  Instances: < AngleAtCenter1 >, < TangentSecant1 >

4.1.7 Operations

This class contains the methods for operation or modification of geometrical shapes of any kind.

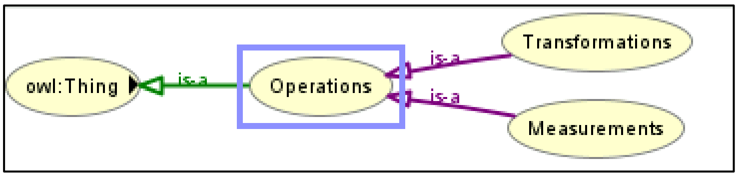


Figure 10: Sub-Classes of Operations

Sub-classes include:

- **Transformations**: Includes Translations, Rotations, Scaling and Reflections.

  Instance: < **Transformations** 1>
- **Measurements:** Includes scenarios such as Area Computation, Volume Computation, and Perimeter Computation.

  Instance: **< Measurement1 >**

4.1.8 Operators

Operators are defined to perform different geometry related operations and geometrical relations.

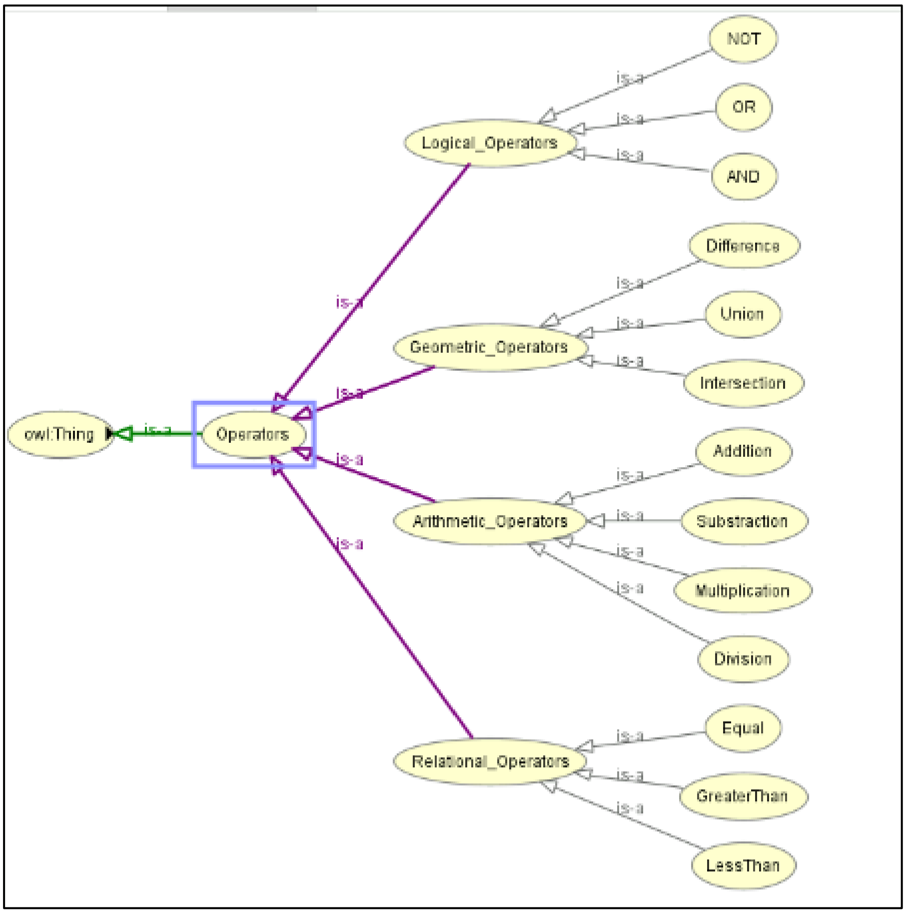


Figure 11: Sub-Class of Operators

- **Arithmetic Operators:** There are four operations for numbers which are Addition, Subtraction, Multiplication and Division.

  Instances: <Addition>, < Subtraction >, < Multiplication>, < Division>

- **Relational Operators:** Enumerations of geometric values or measurements for comparison, MoreThan or LessThan, Equal.

  Instance: < comparison1>
- **Logical Operators:** Perform operations on geometric proofs including And, Or and Not.
  Instance: < LogicalAnd1 >

- **Geometric Operators:** Focus on shape-related operations such as Intersection, Union, and Difference.

  Instance: < Union1 >

4.1.9 Data Type

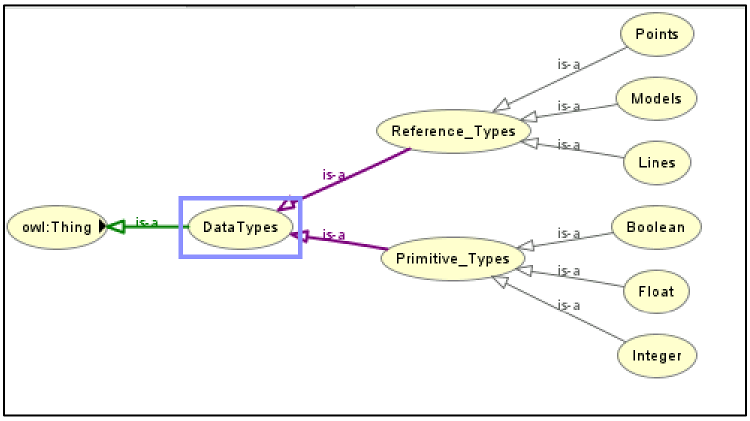


Figure 12: Data Types

1. **Primitive Types**:
   - **Integer**: For discrete counts like the number of sides or vertices. Instance: <**Integer1**>
   - **Float**: For precise measurements such as length, area, or angles. Instance: <**Float1**>
   - **Boolean**: For logical values like whether two shapes intersect. Instance: <**Boolean1**>

2. **Reference Types:**
   - **Points:** Represented as coordinates (x,y,z). Instance: <**Points1**>
   - **Lines:** Defined by equations or endpoints. Instance: <**Lines1**>
   - **Model:** Represented as objects with various attributes. Instance: <**Shapes1**>

4.1.10 Pedagogy Model

The pedagogy or the instructor model can be viewed as the "instructor in the box", which, when equipped with appropriate strategies, can close the gap between students' knowledge and expert geometric knowledge. Within this framework, the instructor module is to define gaps in knowledge in geometry, prioritize these gaps, and define how to present the information. In our system, the instructor model include level, quizzes and feedback. The system follows a structured and progressive learning path categorized into levels. It also integrates quizzes and feedback mechanisms for personalized learning.

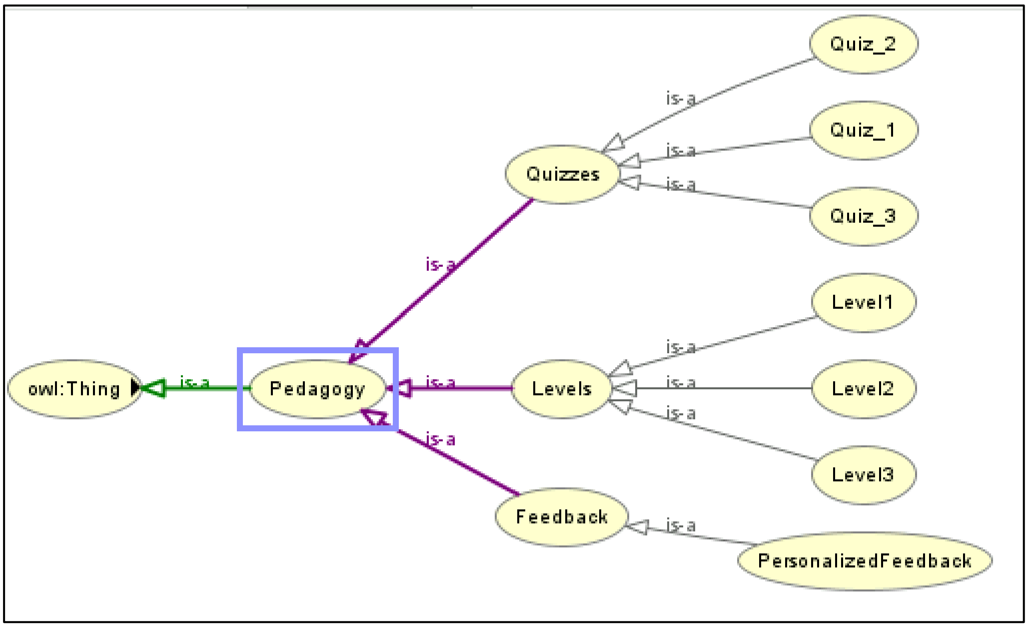


Figure 13: Pedagogy Model

1. **Levels:** The Geometry Contents are categorized into three sub-classes.

    - **Level 1:** It Contain 2D Shapes, Including Triangles (equilateral, isosceles, scalene), rectangles (squares and general rectangles), and circles (radius, diameter, and area).
    - **Level 2:** Focuses on 3D shapes (e.g., spheres, cubes), operations (volume, surface area), Theorems and advanced concepts like geometric operators (intersection, union, difference).

    - **Level 3:** Introduces compound shapes and transformations like translation, rotation, scaling, and reflection.

2. **Quizzes:**

    - **Quiz 1:** Calculate areas of various 2D shapes.
    - **Quiz 2:** Analyze 3D shapes and their properties.
    - **Quiz 3:** Solve problems involving transformations and compound shapes.

3. **Feedback:**

    - **Personalized Feedback:** Provides specific guidance based on the student's quiz performance.

      Example: Encouragement and tips are tailored to motivate and strengthen weak areas**.**

Instances: <Levels 1>, <Quizzes 1>, <Feedback 1>

The following outlines the contents of each level in the geometry domain ontology. The connections between content and levels are represented using the **"BelongsTo"** object property in the ontology structure.

- First level content: This level focuses on foundational geometry concepts

  1. 2D Shapes
     <Rectangle> <BelongsTo> <Level 1>

  2. Angles (Acute, Obtuse, Right)
     <Angle> <BelongsTo> <Level 1>

  3. Area of Simple Shapes.
     <Rectangle> <HasMeasurement> <Area>

  4. Congruence and Similarity
     <Triangle> <IsCongruenTo> < EquilateralTriangle >

- Second level content: This level introduces intermediate geometric concepts

  1. Circles (Chord, Diameter, Tangent)
     <Circle> <HasProperty> <Radius>

  2. Pythagorean Theorem
     <RightTriangle> <Follows> <PythagoreansTheorem>

  3. Volume and Surface Area of Cylinders
     <Cylinder> <HasMeasurement> <Volume>

  4. Trigonometry in Geometry
     <Trianlge> <Uses> <TrignometricFunction>

- Third level contents: The final level deals with advanced topics

  1. Advanced transformations (Scaling and Shearing).
     <Square> <Undergoes> <Scaling>

  2. Geometrical Transformations–Translate, rotate and reflection
     <Trianlge> <Undergoes> <Reflection>

  3. Distance, Midpoint formula and Locus concepts belong to advanced coordinate geometry.
     <Point1> <DistanceFrom> <Point2>

4. Coordinate Geometry Basics (Graphing points, Line formulas)
   <Line> <HasProperty> <Slope>

5. Geometry – Reasoning and Proving
   <Trianlge> <Follows> <AngleSumProperty>

4.1.11 Student Model

The Student Model identifies the current knowledge status of a student, the skills the student possesses, and the goals in the context of geometry. This model is a knowledge structure that represents the student’s path through the ontology. It shows which concepts the student has grasped, which ones the student is currently studying, and what the student wants to accomplish.

In this model, there is the Student class which has sub classes like the Skill and Goal that will help in defining the advancement of a student.

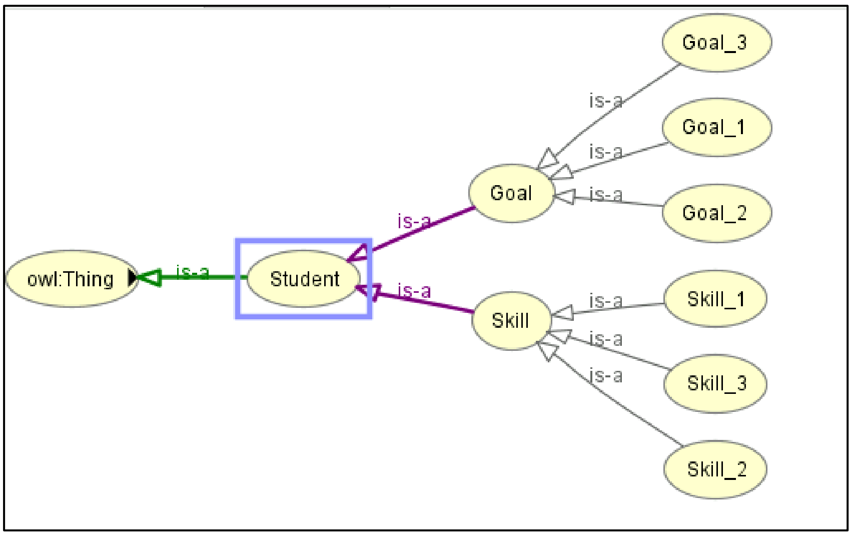


Figure 14: Student Model

The examples of the Student class are real-life examples of students at different levels of learning geometry.

Example1:
Student1:

- **HasCompleted** level 1, which covers foundational geometry concepts, contains 2D Shapes, Including Triangles (equilateral, isosceles, scalene), rectangles (squares and general rectangles), and circles (radius, diameter, and area).
- **IsCompleting** level2, which includes intermediate topics (3D) like polygons and transformation.
- **HasCompleted** Quiz1, a formative assessment of level1 topics.
- **HasSkill** to identify and calculate angles and classify basic shapes (eg: traingles and squares).

Example 2:
Another instance, **Student1**, provides further insight:

- **IsCompleting** Level1, at the moment working on “Base and Height” as one of the topics within the geometry curriculum.
- **HasGoal** to master the concept of congruence and similarity of triangles.
- **ReceivesFeedback** through Feedback1 which is to review more practice problems on perimeter and area.

  <Student> <HasCompleted> <Level 1>
  <Student> <IsCompleting > <Theorem>
  <Student> <HasSkill> <Skill>
  <Feedback> <BelongsTo> <Student>
  <Goal> <IsSetBy> <Student>

4.1.12 Object Properties and Data Properties

Object properties are used to describe the relations between two individuals (instances of classes) in the geometry domain ontology. These properties show the relations of students, concepts, goals and other aspects of a learning system.

Object Properties:

1. AppliesTo:

   - <Theorems> <AppliesTo> <Shapes>

2. BelongsTo:

   - Links a geometry concept (e.g., "TriangleTheorem") to the level it belongs to.
   - <TriangleTheroem> <BelongsTo> <Level 1>

3. HasCompleted:

   - Relates a student to a level, quiz, or skill they have completed.
   - <Student1> <HasCompleted> <Quiz1>

4. IsCompleting:

   - Represents a student currently working on a level or concept.
   - <Student1> <IsCompleting> <Circle/Operations>

5. ReceivesFeedback:

   - Associates feedback with the student.

- <Feedback1> <BelongsTo> <Student1>

6. HasGoal:

    - Links a student to their learning objectives.
    - <Student1> <HasGoal> <Goal1>

7. HasSkill:

    - Indicates the skills a student has acquired.
    - <Student2> <HasSkill> <CanCalculateTriangleArea>

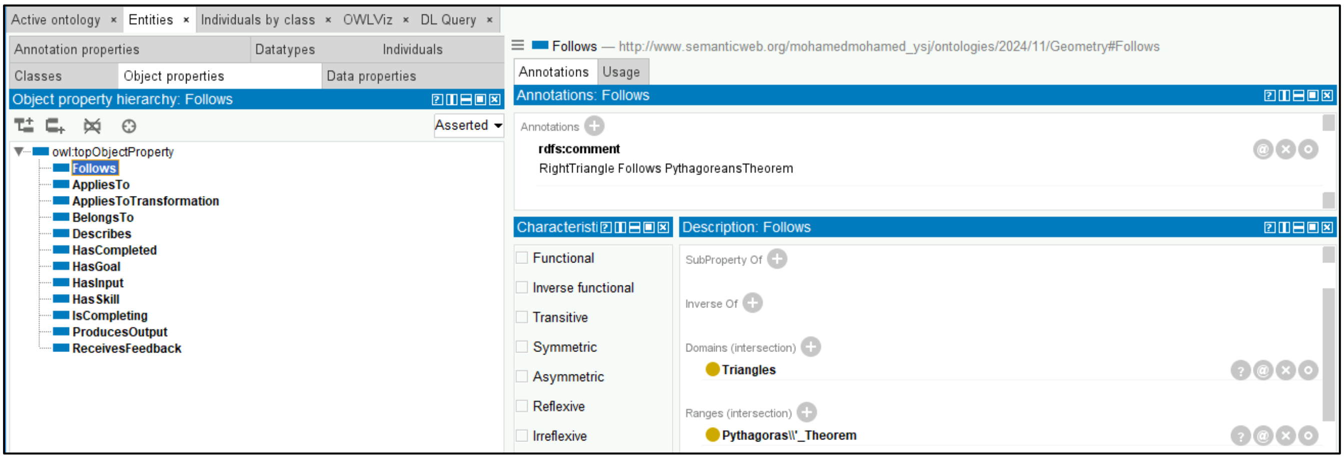


Figure 15: Object Properties

Data Properties:

1. HasName:

    - Stores the name of a student or geometry concept.
    - <Student1> <HasName> <"Nusly">

2. HasLevelName:

    - Defines the name or title of a level.
    - <Level 1> <HasLevelName> <2D>

3. HasScore:

    - Represents a student's score in a quiz or assessment.
    - <Student3> <HasScore> <85>

4. HasProgress:

    - Tracks the percentage of completion of a level or concept.
    - <Student1> <HasProgress> <60%>

5. HasGoalDescription:

- Describes a student's learning goal in detail.
- <Goal1> <HasGoalDescription> <Master congruence and similarity>

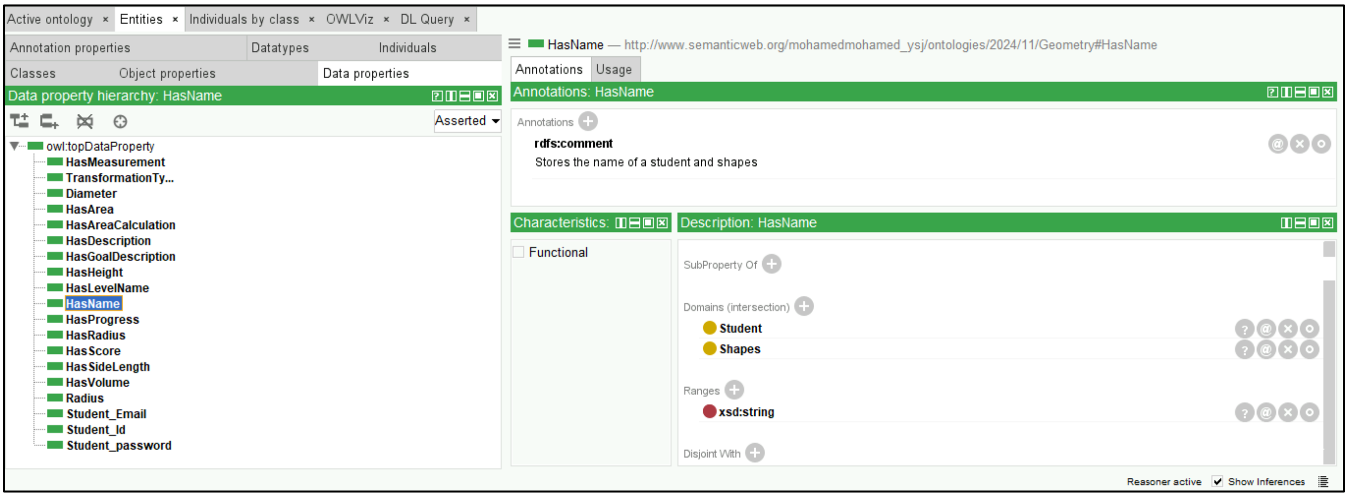


Figure 16: Data Properties

This is the snapshot of Instance create.

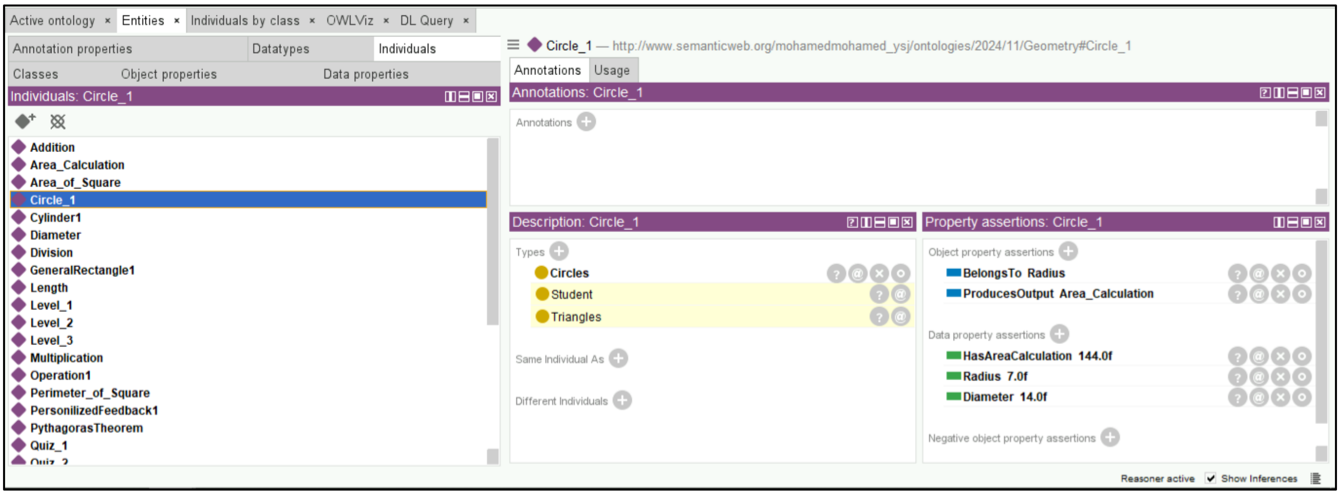


Figure 17: Individuals

4.2 Designing with Figma

The created ontology was modeled by designing user interfaces for a mobile app to teach geometry using the tool ‘Figma’. Nomadic: These are interfaces developed for a learner to practice geometrical concepts such as calculating areas of shapes. The design shows how the main three classes, their sub-classes, data properties and object properties of the ontology are arranged. These details will be discussed in more detail in the subsequent sections.

First, loading page was created as shown in figure with an appropriate logo.

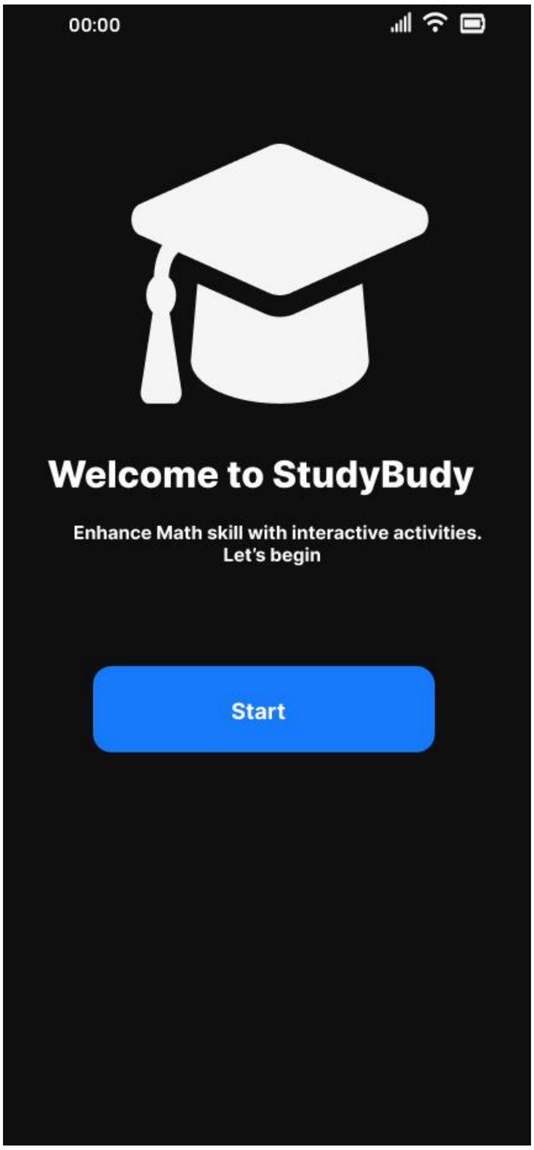


Figure 18: Loading Page

The app is intended for use by learners of all ages. New users can register to create an account while the existing users can directly login to start their learning process.

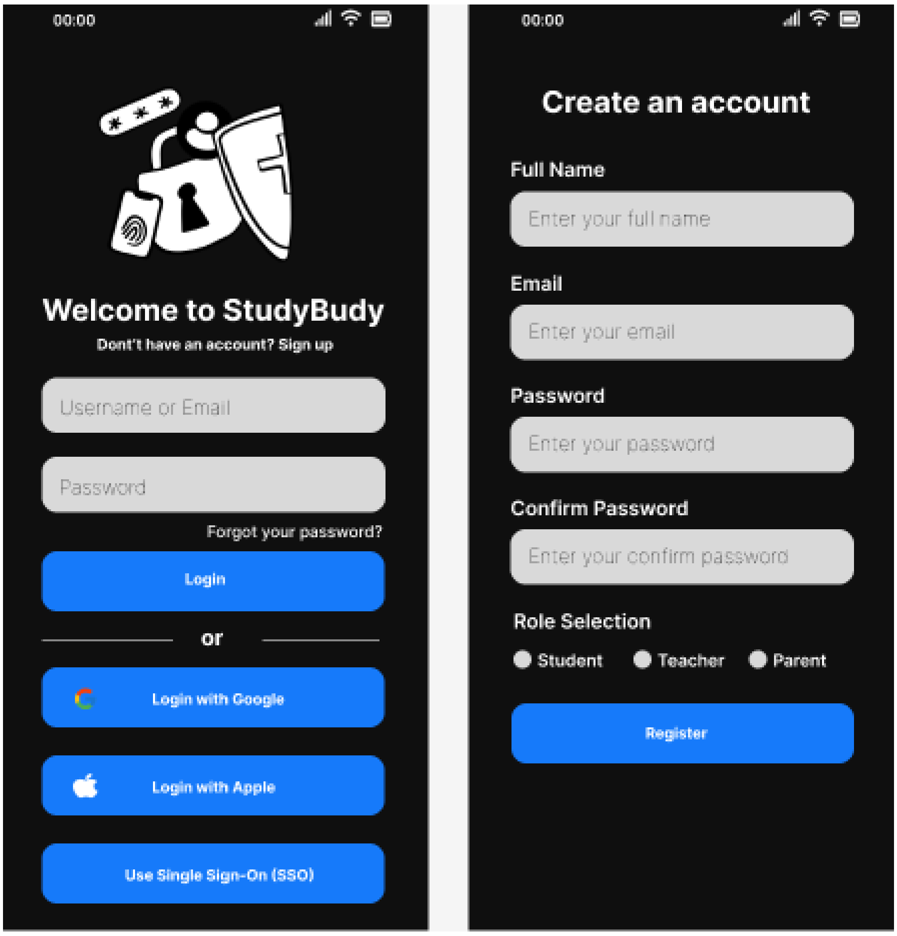


Figure 19: Login/Signup Page

Once a user sign-up/login, the user will be directed into the 'Home Page' as shown in figure 18. The home page is thus the first point of access to learning geometry. The resource is designed with fun and colorful statistics to capture the attention of the learner. The upper most section of the page has a concise tagline addressing the user and welcoming them. Bottom of the page consists of different icons for easy navigation. The parts of the bottom navigation line are Home, Lessons, Search, Message, and Account.

Middle section provides information regarding the learner making it easy for them to track their learning time, next lessons, as well as overall lesson progress at one glance. When a learner clicks ‘Click for more’ learner will be able to see their progress in detail with exact numbers.

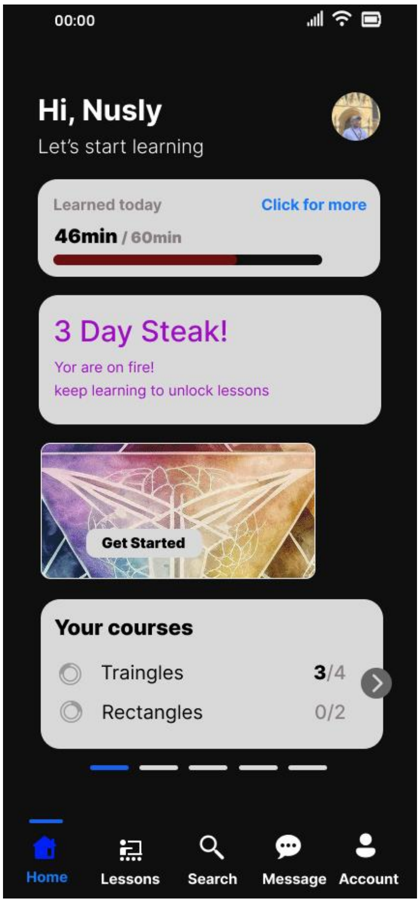


Figure 20: Home Page

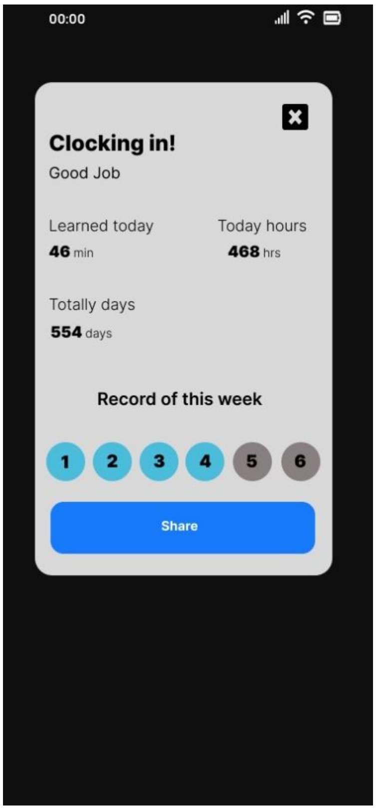


Figure 21: 'Click for more' page

Figure 20 shows a screenshot of a “Lessons” part of a learning application. It presents a list of topics and categories for learning. Above the list of lessons, there is a bar for searching lessons, and below the list, there is a section called “My Lessons.” Below that, there are lesson topics such as Shapes, Theorems, Operations, and others, each with the number of topics and a “Click” button to access them. It has a simple graphical interface that is also bright as compared to other RSS feed readers. On the bottom, there are Home, Lessons, Search, Messages, and Account buttons.

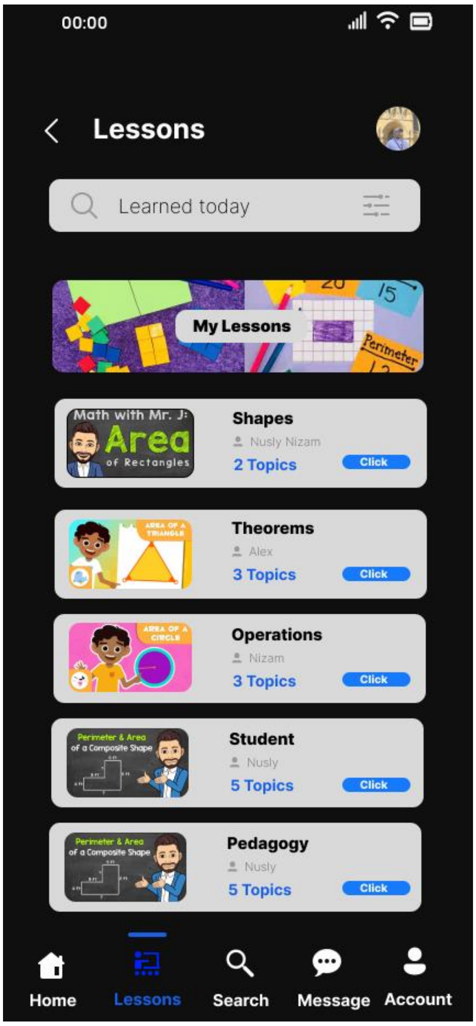


Figure 22: 'Lessons' page

After navigating into the 'My Lessons' page, the content is divided according to levels. The relevant quizes of the levels will be locked until all the lessons of the level is completed.

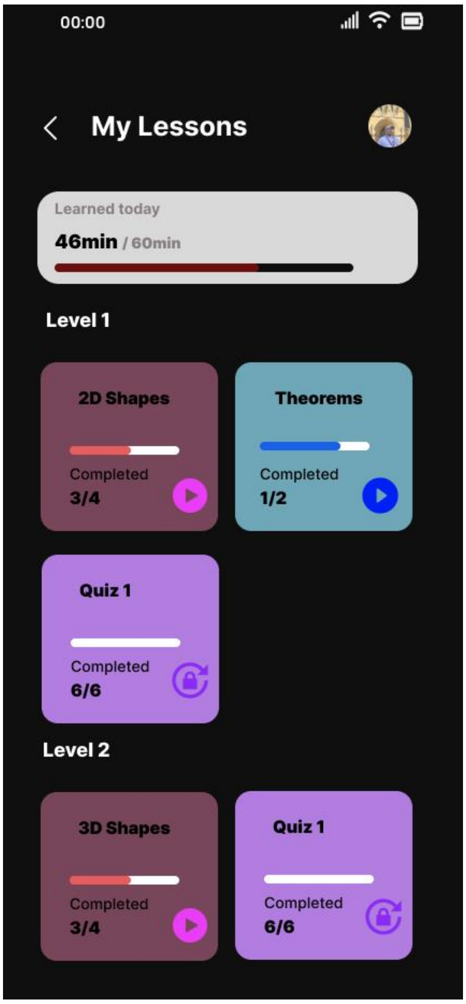


Figure 23: 'My Lesson' page

The lessons in the application are divided into levels and topics. Every lesson has several topics and a learner is required to cover all the topics in a given level before moving to the next level. In that image, the learner has one unit out of four in 2D Shapes and half of one unit of Level 1 Theorems. They have also done Quiz 1 in Level 1 and are still doing 3D Shapes in Level 2. This is in line with the ontology model in which lessons are categorized into topics and subtopics and each topic is given a level.

The following figure 22 shows a “Shapes” category in a learning application. There are two lessons: 2D shapes and 3D shapes, each of which has four topics. It also shows the duration of lessons: “4 days” for 2D Shapes and “10 hours” for 3D Shapes. It is convenient to search and monitor. Additionally when a basic learner click the topic of **2D Shapes,** then this will be navigate to another page.

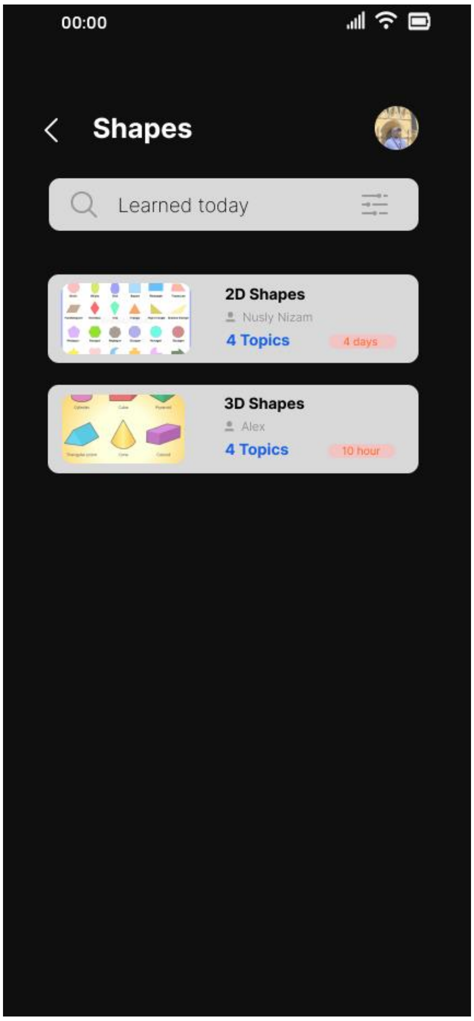


Figure 24: 'Shapes' page

This page is sub-class of 2D Shapes. This screenshot illustrates the 2D Shapes lesson page in a learning application. It includes such topics as Rectangles, Triangles, Circles, Compound Shapes, etc., with the number of topics and estimated time, for instance, 1 hour. These are tabs such as All, Favourites, and Completed to ease the tracking of progress. It’s simple and engaging!.

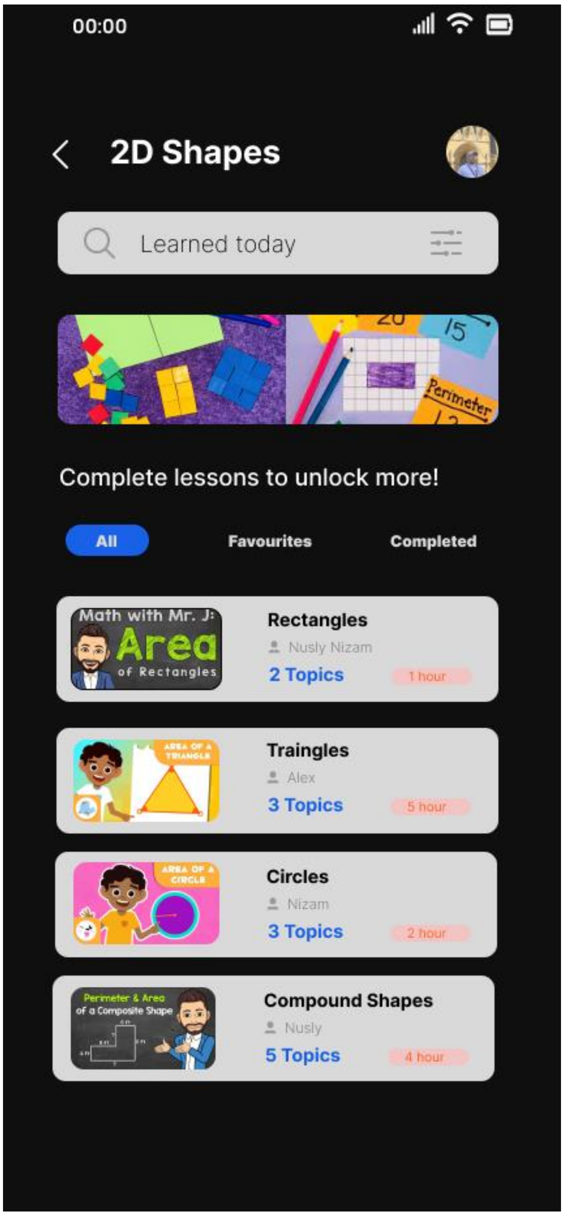


Figure 25: '2D' Shapes' page

The figure 24 shows the sub-class of Rectangles. And its includes list of videos tutorials. The videos are categorized into two types: "Square" and "General Rectangle".

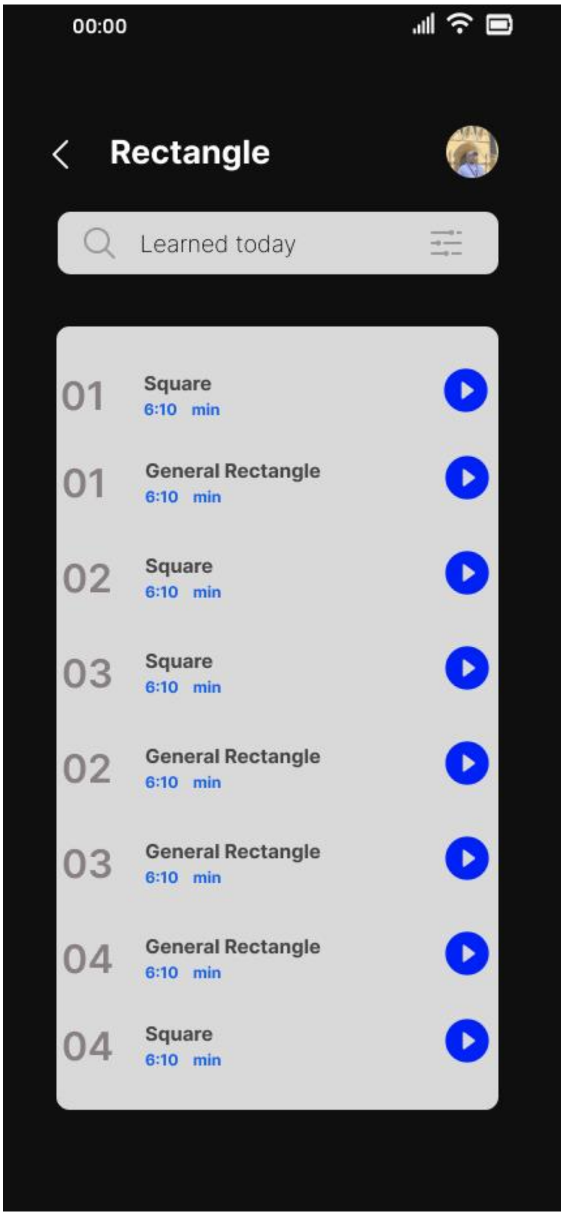


Figure 26: 'Rectangle' page

Figure 25 shows the interface of individual lessons in 2 instances. The left-hand instance shows the default view of the interface while the right instance shows the interface when the preview of the lesson is playing. When a learner clicks on a lesson, the learner will be directed to the left user interface. In the lesson, one can see all the topics and subtopics as a list. A preview of the topic is provided. Unless a learner completes the topics related to the specific level, the learner cannot go to the next topics in the next level. They will be locked. They can click on ‘Start Learning’ to start the lessons and can add the lessons to their favourite lessons list through clicking star.

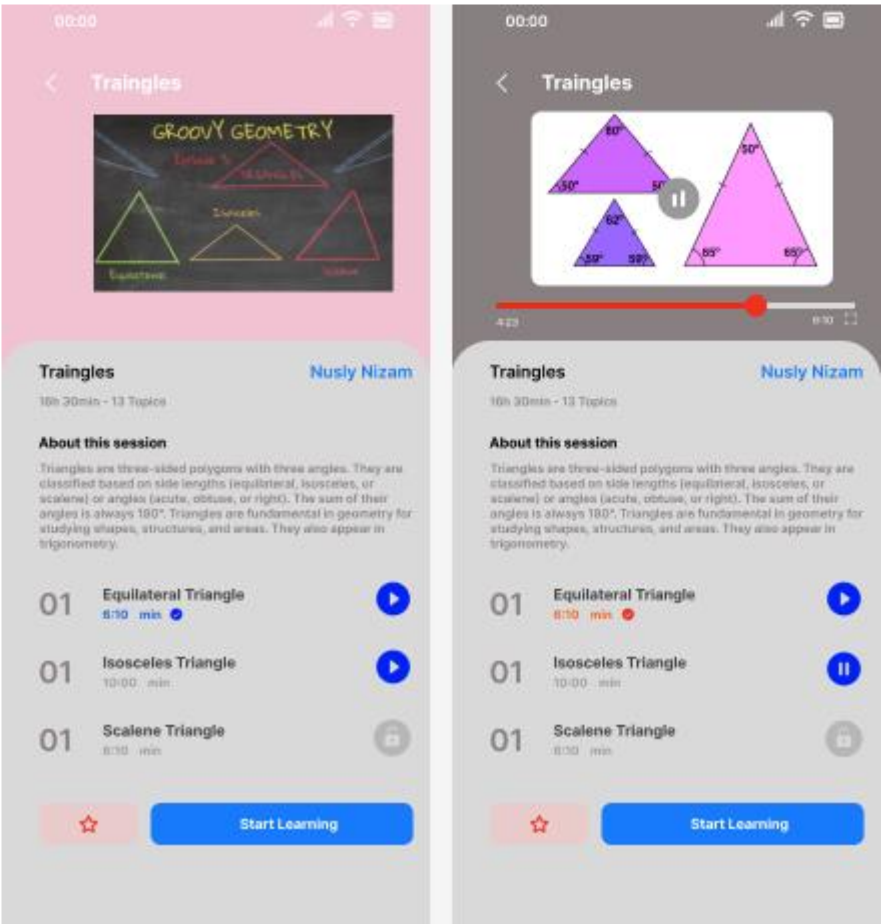


Figure 27: 'Triangles' pages

This picture (Figure 26) illustrates the search results of the word “Triangle” in a learning application. The results include videos on the various categories of triangles such as equilateral, isosceles, scalene among others. There is also a section on triangle theorems

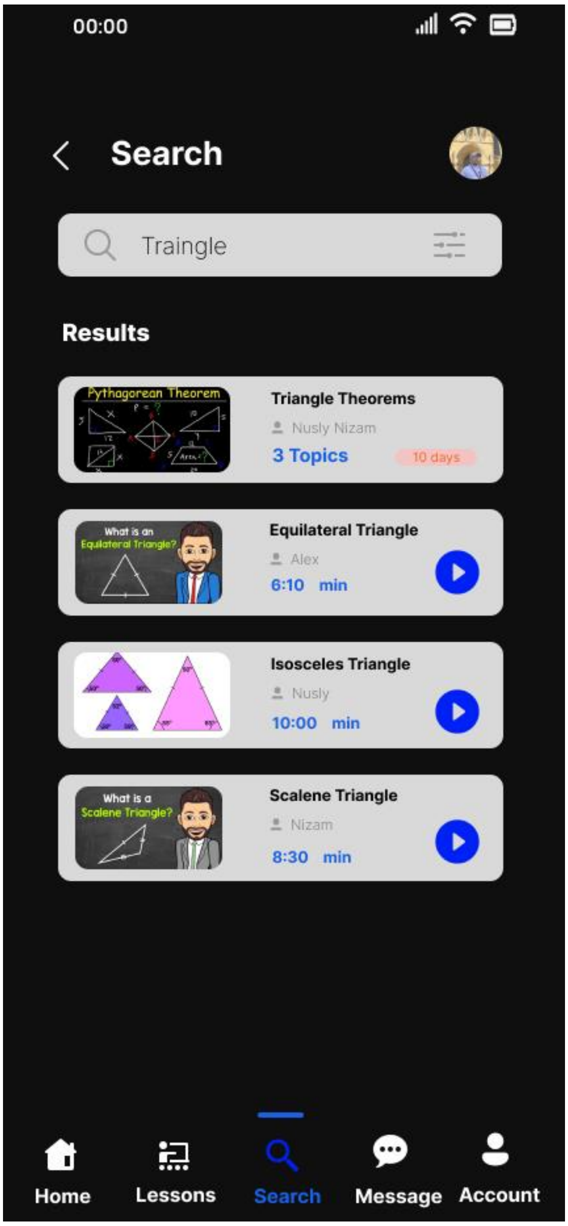


Figure 28: 'Search' page

An advanced filter has been added as shown in figure 27 to the search bar for easier access to the necessary content. This image illustrates a search filter in a learning application. It is possible to sort the videos by categories (shapes, theorems, operations), level (from 1 to 3), length of the video, and 2D/3D. You can also remove the filters or use it and to see the results with the given filters.

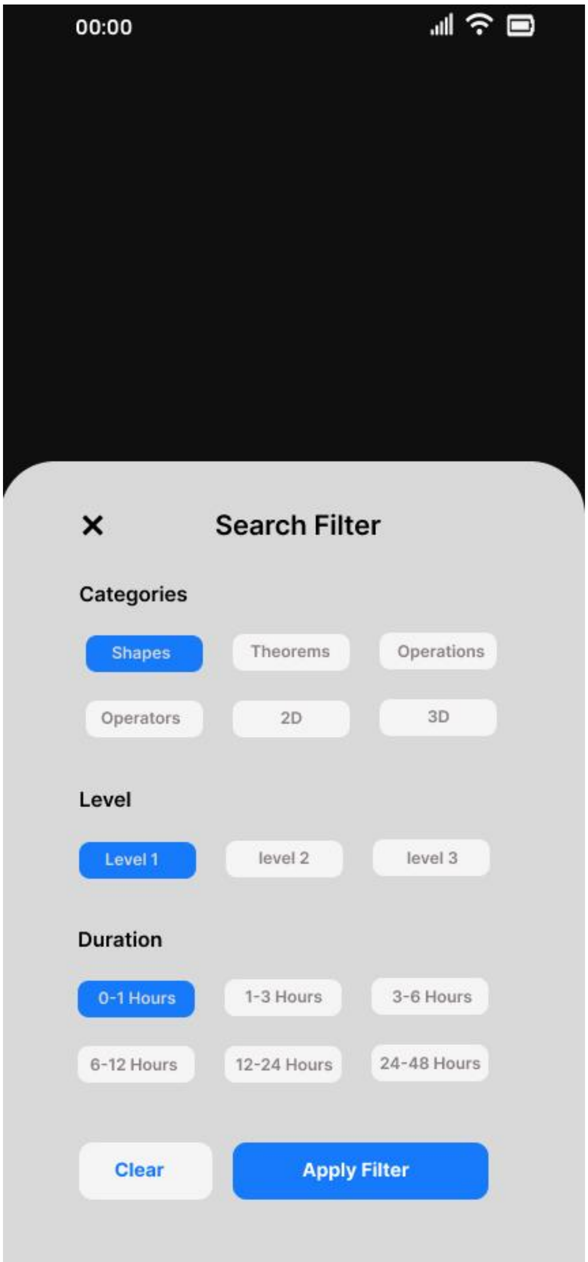


Figure 29: 'Search Filter' page

This image (figure 28) illustrates the account page of a learning application. Here you can view your profile picture, change your account details, set your goals, control your privacy, and find the help. The options are Home, Lessons, Search, Messages, and Account are displayed at the bottom of the page in the navigation bar. ‘Edit Account’, ‘Settings and Privacy’, ‘Skills and Goals’ and ‘Help’ options.

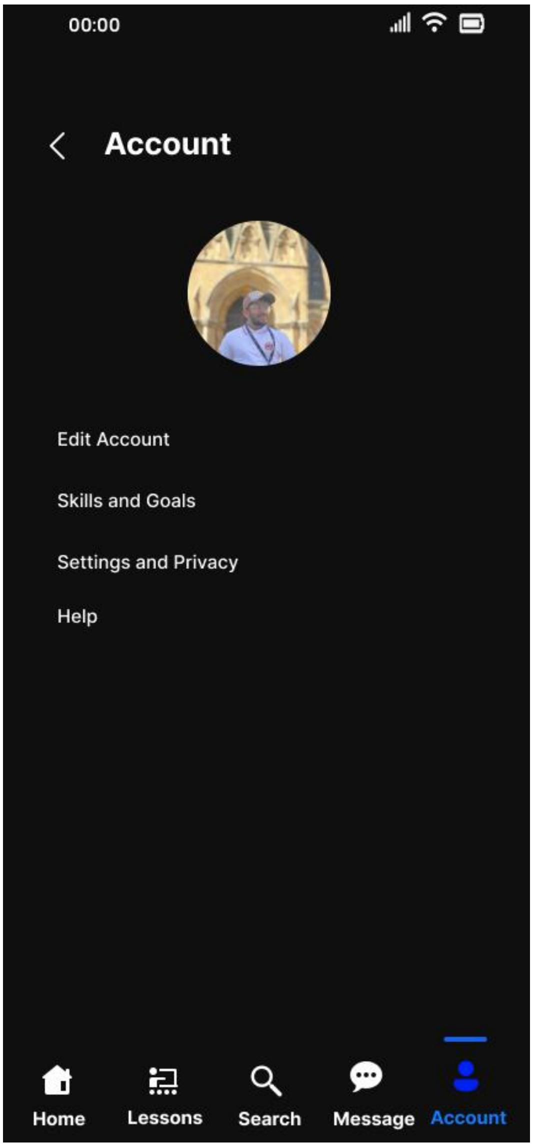


Figure 30: 'Account' page

In the ‘Skills and Goals’ section, the predefined skills and goals can be seen as shown in figure 29. The percentage of completed goals are shown along with the unlocked skills. Incomplete skills related to incomplete lessons are shown in grey colour with a lock.

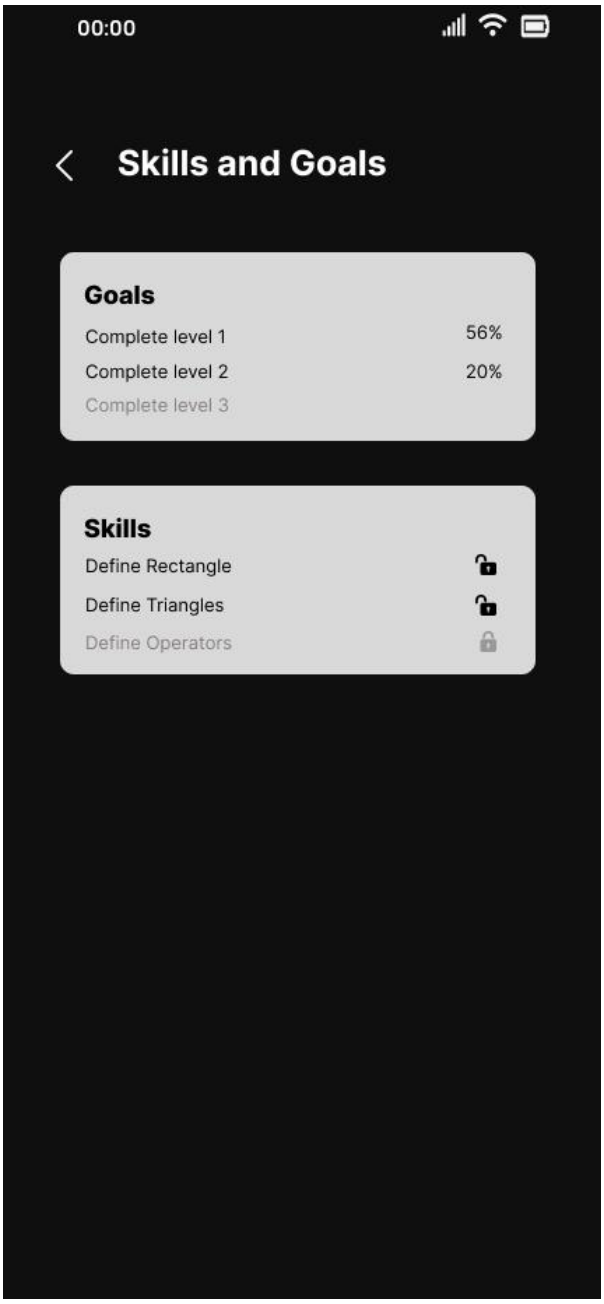


Figure 31: 'Skills and Goals' page

This image (fig 30) illustrates the message part of a learning application. You can view the latest messages from other users, for example, congratulating on the completion of a lesson or on changes in the course. The messages include the name of the sender, the profile picture, the time of the message and the status of the sender.

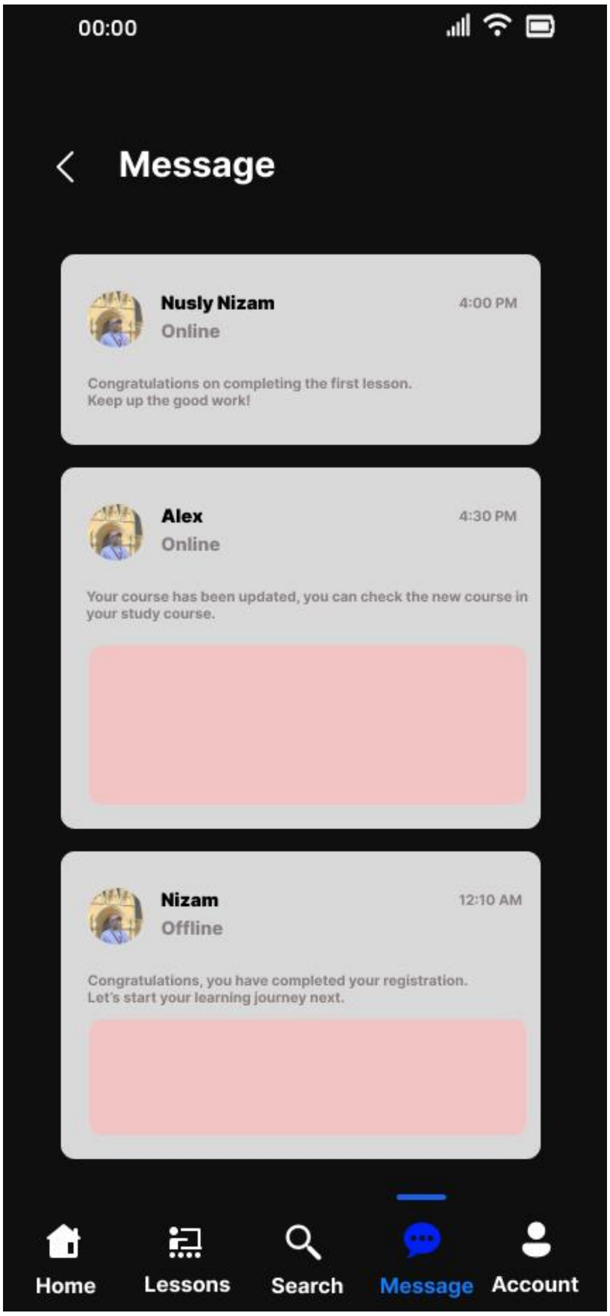


Figure 32: 'Message' page

4.3 System Integration

In this section additionally, I have created some of functional User Interfaces in Python code according to my system and how it will connect with my ontology.

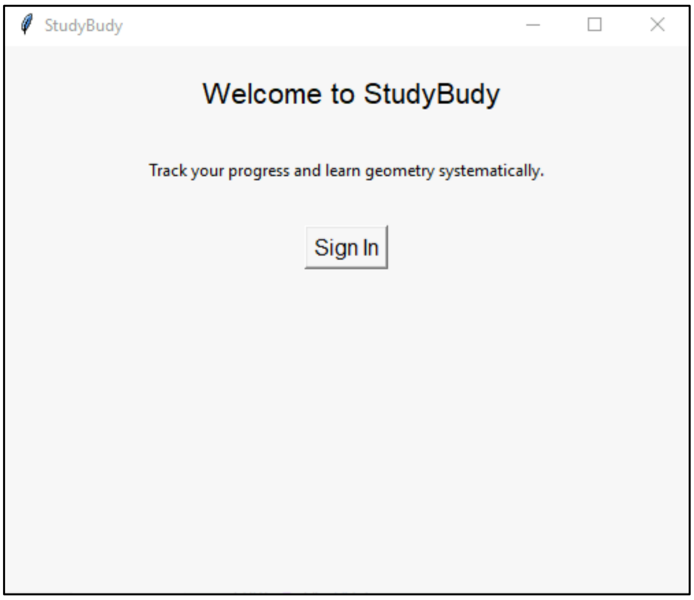


Figure 33: Welcome Screen

Figure 34: SignIn Page

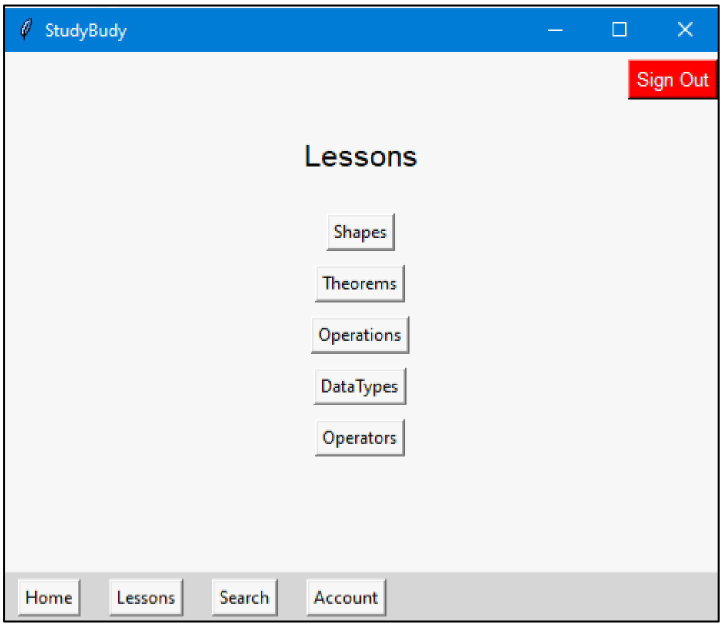


Figure 35: Lesson Page

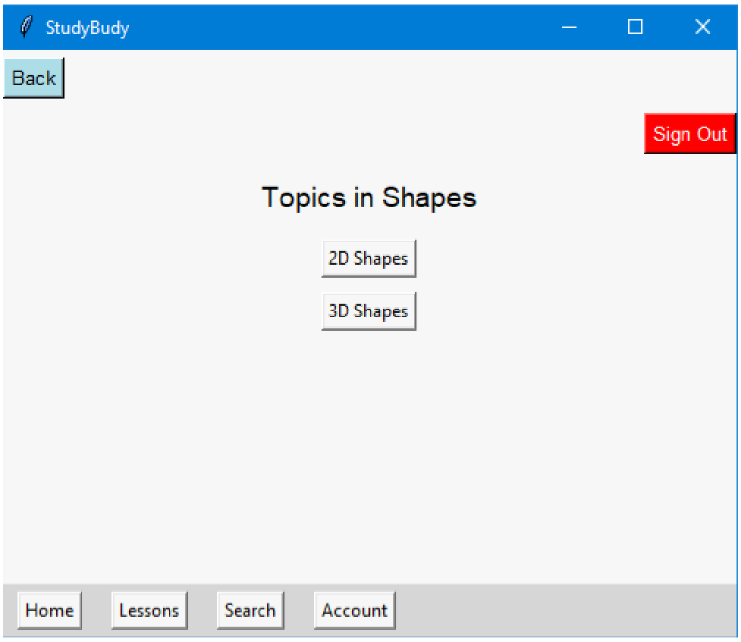


Figure 36: Shapes Page

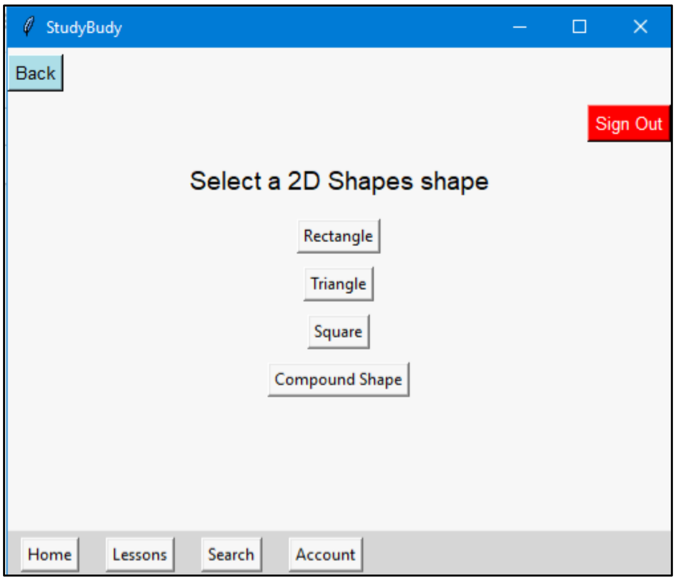


Figure 37: 2D Shapes

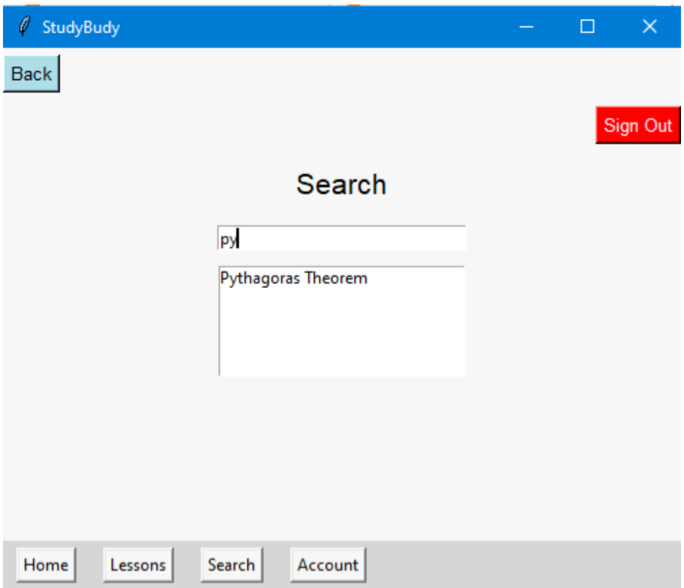


Figure 38: Search Page

Search page is used to search Geometry content. Specially this page was designed SEO pattern. Its mean when a user type a letter to search, it automatically gives the suggested list in the list box.

Furthermore, According to these user interface, owlready2 query is used to fetch the data according to our ontology creation. (eg: checking the number of classes, object properties, data properties and instances, etc.)

Finding Object Property Relationships

```
[59]: from owlready2 import *

      # Load the ontology
      onto = get_ontology("ITS (Geometry) ontology.owl").load()

      # Retrieve all object properties (relationships between individuals)
      print("Object property relationships between individuals:")
      for prop in onto.object_properties():  # Iterating only over object properties
          for subj in onto.individuals():  # For each individual
              if getattr(subj, prop.name):  # Check if the individual has this property
                  for obj in getattr(subj, prop.name):  # Iterate through the related individuals (if any)
                      print(f"{subj} --{prop}--> {obj}")
```

```
Object property relationships between individuals:
Geometry.PythagorasTheorem --Geometry.AppliesTo--> Geometry.RightTriangle
Geometry.Student_2 --Geometry.AppliesTo--> Geometry.Quiz_2
Geometry.Area_of_Square --Geometry.AppliesToTransformation--> Geometry.Square
Geometry.Circle_1 --Geometry.BelongsTo--> Geometry.Radius
Geometry.GeneralRectangle1 --Geometry.BelongsTo--> Geometry.Rectangle1
Geometry.Rectangle1 --Geometry.BelongsTo--> Geometry.Level_1
Geometry.Student_1 --Geometry.BelongsTo--> Geometry.Quiz_2
Geometry.Area_Calculation --Geometry.Describes--> Geometry.Circle_1
Geometry.RightTriangle --Geometry.Follows--> Geometry.PythagorasTheorem
```

Figure 39: Find Object properties relationship between individuals

Retrieve All Data Properties

```
[50]: from owlready2 import *

      # Load the ontology
      onto = get_ontology("ITS (Geometry) ontology.owl").load()

      # Retrieve all data properties
      print("Data Properties in the ontology:")
      for prop in onto.data_properties():
          print(prop)
```

```
Data Properties in the ontology:
Geometry.Diameter
Geometry.HasArea
Geometry.HasAreaCalculation
Geometry.HasDescription
Geometry.HasGoalDescription
Geometry.HasHeight
Geometry.HasLevelName
Geometry.HasMeasurement
Geometry.HasName
Geometry.HasProgress
Geometry.HasRadius
Geometry.HasScore
```

Figure 40: Retrieve all data properties

5 Conclusion

Therefore, ITS means an innovative model for learning in certain subjects like geometrics because an ITS considers individual difference. This research describes the development of an ITS for teaching geometry concepts including finding the areas of triangles, rectangles, and circles. By utilizing such elements of artificial intelligence as adaptive learning algorithms and natural language processing the system can make learners’ experience exclusive: it assists them to advance through the material as fast as they are ready for, and to identify the aspects they fail to grasp. Also, by using the ontology-based knowledge representation with the help of tools like Protégé and OWL, the ITS breaks a large amount of information into small lessons, making geometry less scary for learners. Such as Classes, sub-classes, object properties, data properties and instances (individuals). With this in mind, I developed an interactive user-friendly interface with Figma to create engagement and structure the learning for the user. This project demonstrates the possibility of ITS to address the conventional difficulties of teaching geometry by integrating individualization, ease, and technology. It shows how current AI technologies can be used to enhance learning processes, increase their effectiveness and relevance to learners – for mathematics education and other fields.

6 References

Agrawal, 2022. *Agile methodology: Incremental and iterative way of development.* [Online]
Available at: https://medium.com/@ashutoshagrawal1010/agile-methodology-incremental-and-iterative-way-of-development-a6614116ae68

Al-Bastami & Naser, 1970. *design and development of an intelligent tutoring system for C# language.* [Online]
Available at: https://philpapers.org/rec/ALBDAD-4

Antoniou & Harmelen, 1970. *Web ontology language: OWL.* [Online]
Available at: https://link.springer.com/chapter/10.1007/978-3-540-24750-0_4

Corbett, 2001. Cognitive Computer Tutors: Solving the two-sigma problem. *Lecture Notes in Computer Science,* pp. 137-147.

Crow, T., Luxton-Reilly, A. and Wünsche, B.C., 2018. *Intelligent tutoring systems for programming education: a systematic review.* Brisbane, Association for Computing Machinery, pp. 53-62.

Evans, J. E., 2024. The Effectiveness of ALEKS to Address Grade 8 Students' Math Deficiencies. pp. 1-20.

Kularbphettong, K., Kedsiribut,P. and Roonrakwit, P., 2015. Developing an Adaptive Web-based Intelligent Tutoring System Using Mastery Learning Technique. *Procedia - Social and ,* pp. 686-691.

M. Chang, G. D'Aniello, M. Gaeta, F. Orciuoli, D. Sampson and C. Simonelli, 2020. Building Ontology-Driven Tutoring Models for Intelligent Tutoring Systems Using Data Mining. *IEEE Access,* pp. pp. 48151-48162.

M.M El-Khouly, B.H Far and Z Koono, 2000. Expert tutoring system for teaching computer programming languages. *ScienceDirect,* pp. 27-32.

Panagiotopoulos et al., 1970. *An ontology-based model for student representation in intelligent tutoring systems for Distance Learning.* [Online]
Available at: https://link.springer.com/chapter/10.1007/978-3-642-33409-2_31

Peter Brusilovsky, Elmar Schwarz, and Gerhard Weber, 2023. *Elm-art: An intelligent tutoring system on World wide web.* [Online]
Available at: https://rd.springer.com/chapter/10.1007/3-540-61327-7_123

Peter Brusilovsky; Elmar W. Schwarz;Gerhard Weber, 1996. *ELM-ART: An intelligent tutoring system on world wide web.* Montreal, Third International Conference.

Raghu Raman; Prema Nedungadi, 2010. *Performance improvements in schools with Adaptive Learning and Assessment.* San Juan, PR, IEEE, pp. 1-5.

Rekhawi, H.A.A. and Naser, S.S.A. , 2018. An Intelligent Tutoring System for Learning Android Applications UI Development. *International Journal of Engineering and Information Systems (IJEAIS) ,* pp. 1-14.

Sharma, P. and Harkishan, M., 2022. Designing an intelligent tutoring system for computer programing in the Pacific. *Springer Nature Link,* p. pp 6197–6209.

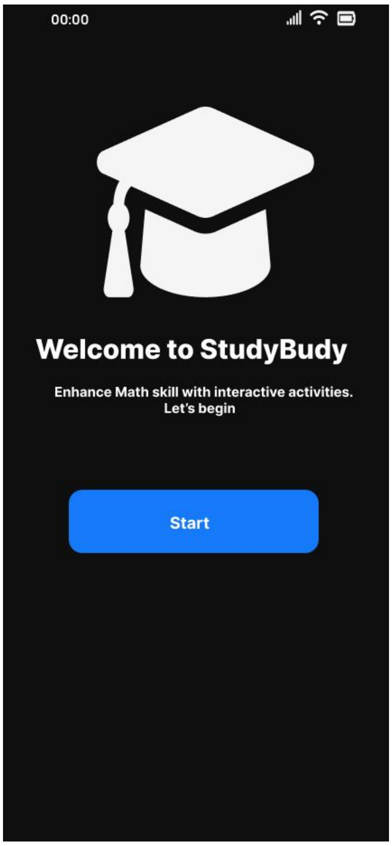
Slawek-Polczynska, 2020. *SolDevelo.* [Online]
Available at: https://soldevelo.com/blog/is-agile-always-the-best-solution-for-software-development-projects/
[Accessed 07 Dec 2024].

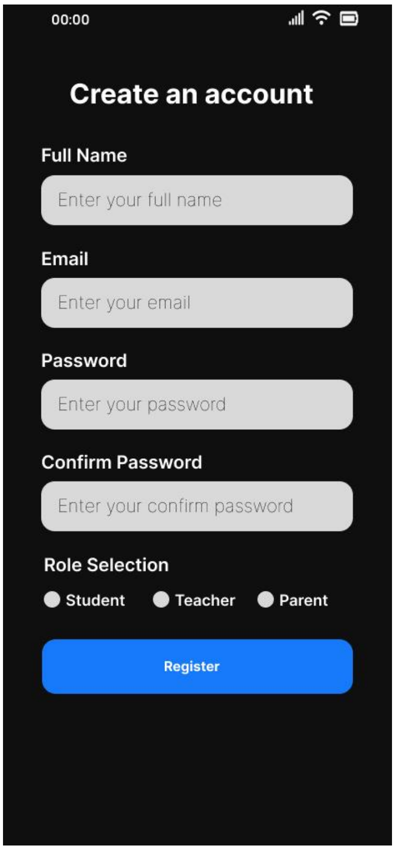
Stankov, S., Glavinic, V. and Rosic, M., 2010. *Intelligent Tutoring Systems in E-Learning.* New York: IGI Global..

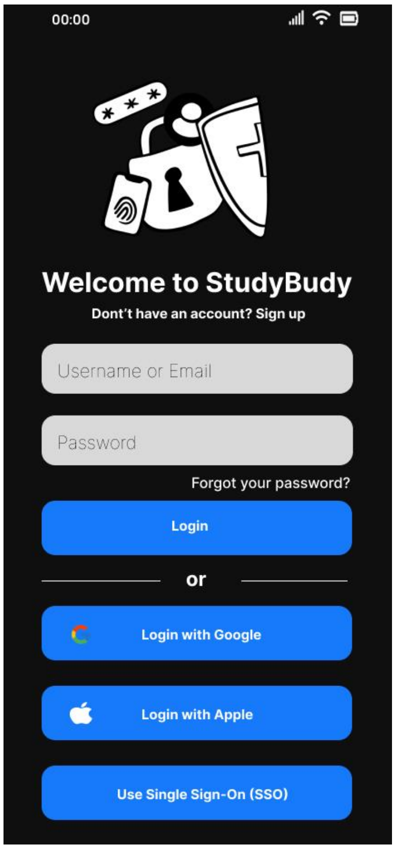
Velichova, 2011. Interactive Maths with GeoGebra. *International Journal of Emerging Technologies in Learning (iJET),* pp. 31-35.

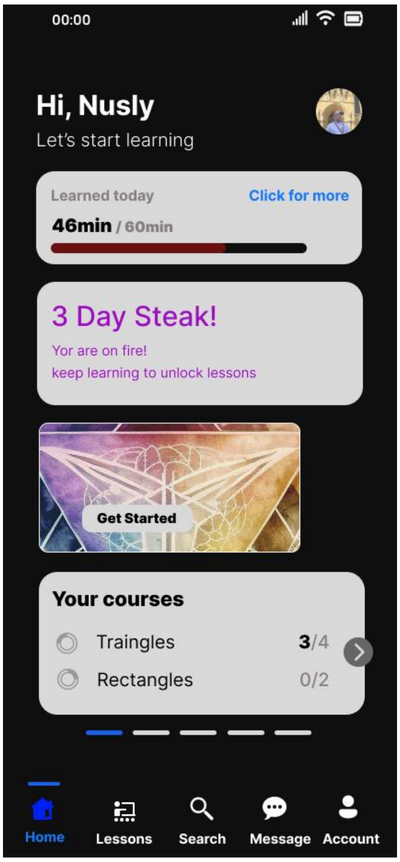
Yılmaz, F.G.K. and Yılmaz, R., 2023. The effect of generative artificial intelligence (AI)-based tool use on students' computational thinking skills, programming self-efficacy and motivation. *Computers and Education: Artificial Intelligence*, p. 100147.

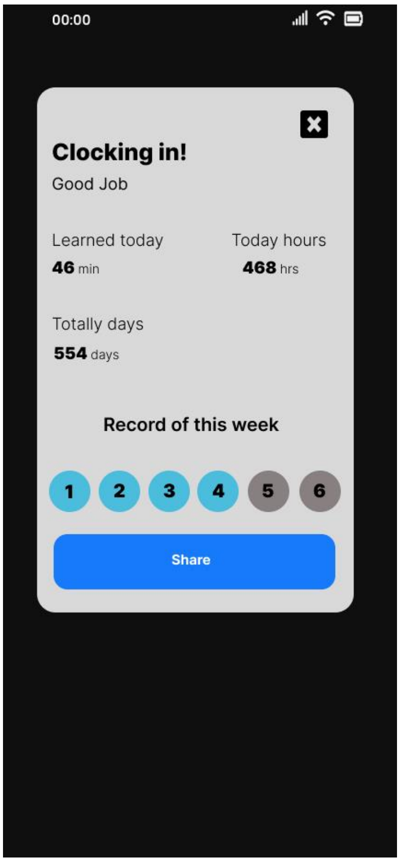
7 Appendix A – User Interface Design

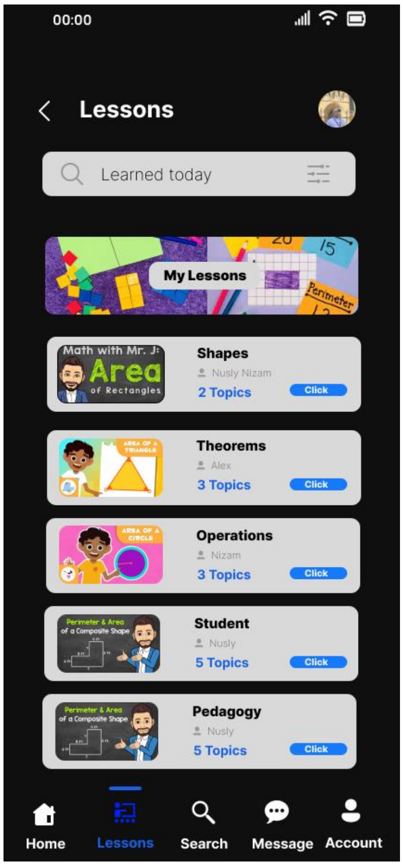


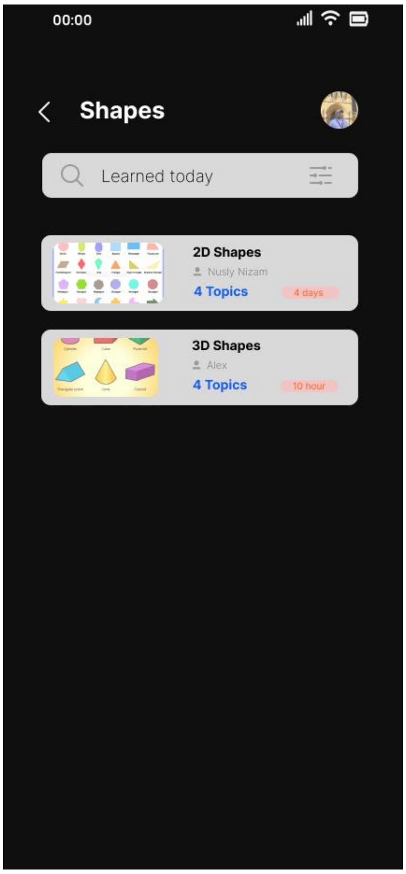


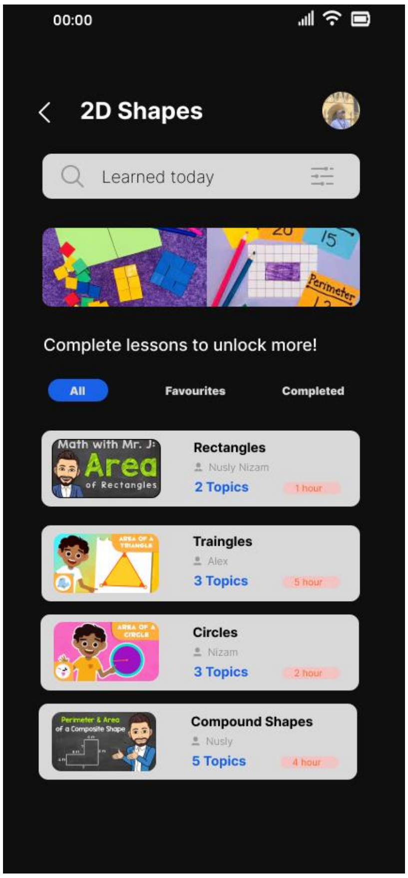


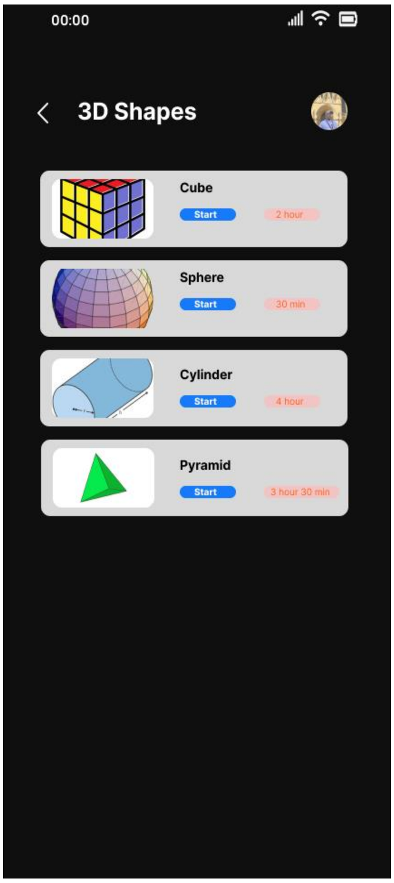


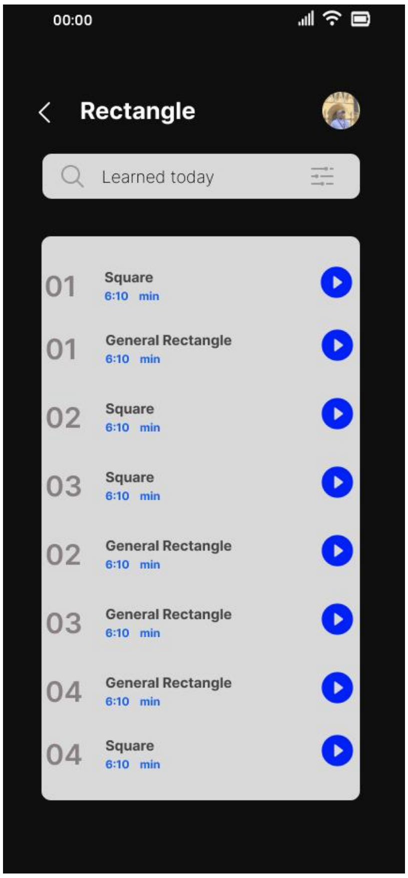


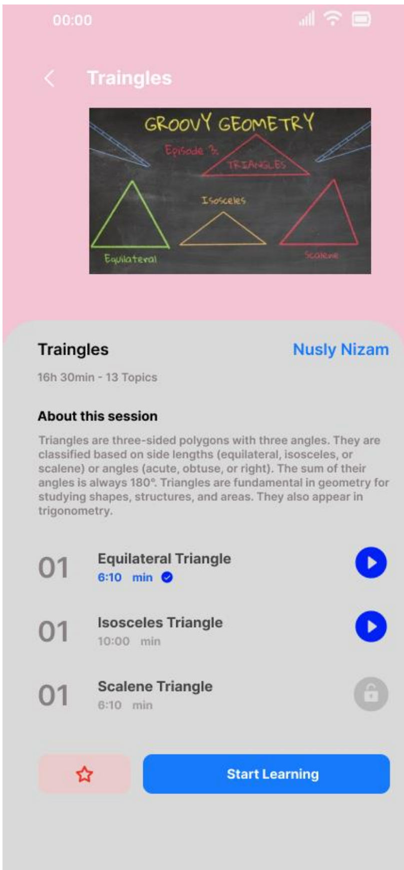


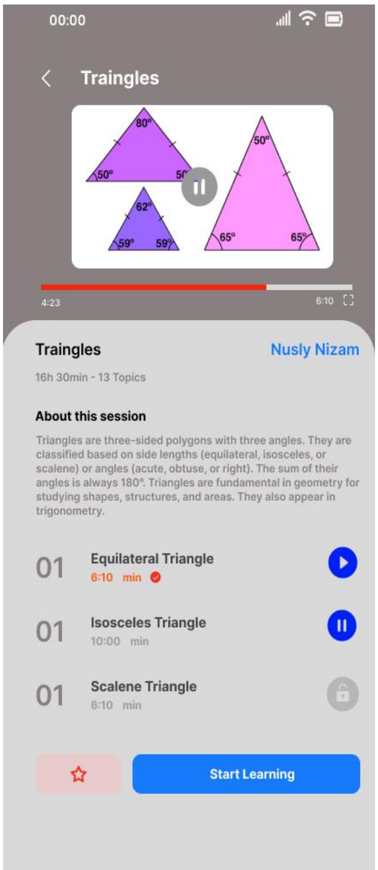

00:00
Shapes
Learned today
2D Shapes
Nusly Nizam
4 Topics
4 days
3D Shapes
Alex
4 Topics
10 hour

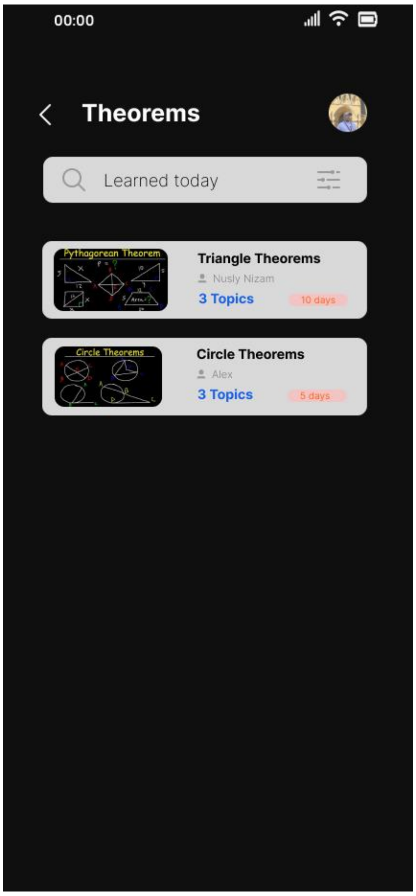

00:00
2D Shapes
Learned today
Complete lessons to unlock more!
All
Favourites
Completed
Rectangles
Nusly Nizam
2 Topics
1 hour
Traingles
Alex
3 Topics
5 hour
Circles
Nizam
3 Topics
2 hour
Compound Shapes
Nusly
5 Topics
4 hour

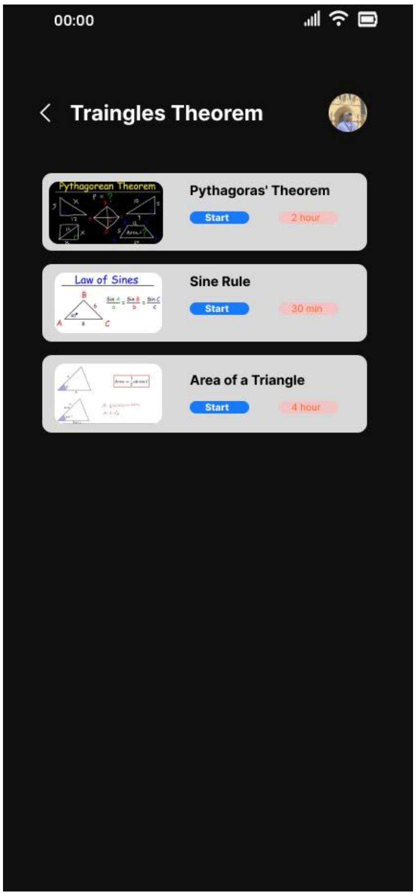

00:00
3D Shapes
Cube
Start
2 hour
Sphere
Start
30 min
Cylinder
Start
4 hour
Pyramid
Start
3 hour 30 min

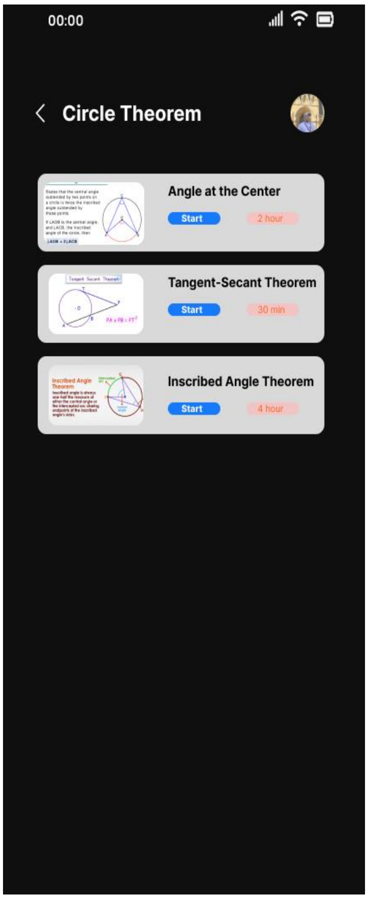

00:00
Rectangle
Learned today
01 Square 6:10 min
01 General Rectangle 6:10 min
02 Square 6:10 min
03 Square 6:10 min
02 General Rectangle 6:10 min
03 General Rectangle 6:10 min
04 General Rectangle 6:10 min
04 Square 6:10 min

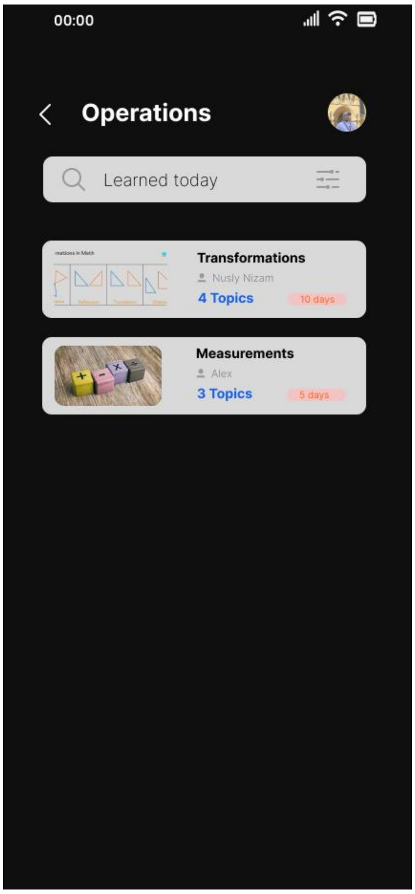

00:00
Traingles
GROOVY GEOMETRY
Traingles
Nusly Nizam
16h 30min - 13 Topics
About this session
Triangles are three-sided polygons with three angles. They are classified based on side lengths (equilateral, isosceles, or scalene) or angles (acute, obtuse, or right). The sum of their angles is always 180°. Triangles are fundamental in geometry for studying shapes, structures, and areas. They also appear in trigonometry.
01 Equilateral Triangle 6:10 min
01 Isosceles Triangle 10:00 min
01 Scalene Triangle 6:10 min
Start Learning

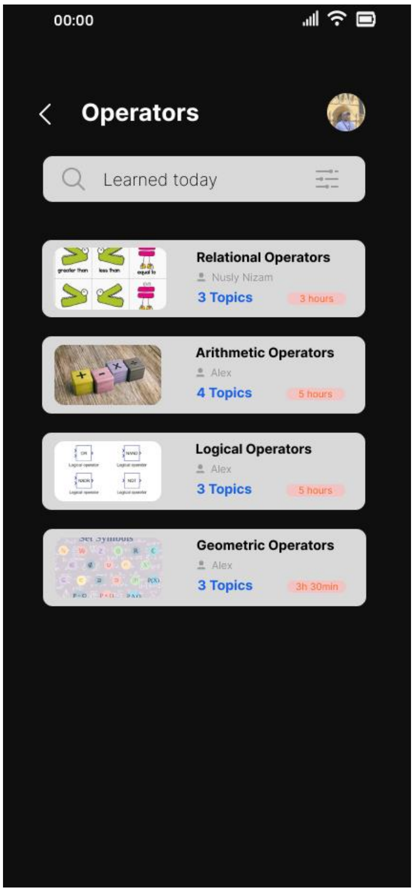

00:00
Traingles
4:23
6:10
Traingles
Nusly Nizam
16h 30min - 13 Topics
About this session
Triangles are three-sided polygons with three angles. They are classified based on side lengths (equilateral, isosceles, or scalene) or angles (acute, obtuse, or right). The sum of their angles is always 180°. Triangles are fundamental in geometry for studying shapes, structures, and areas. They also appear in trigonometry.
01 Equilateral Triangle 6:10 min
01 Isosceles Triangle 10:00 min
01 Scalene Triangle 6:10 min
Start Learning

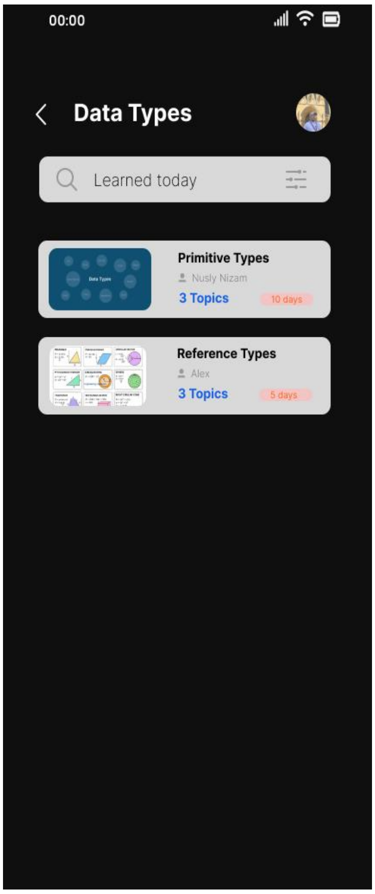

00:00
Theorems
Learned today
Triangle Theorems
Nusly Nizam
3 Topics
10 days
Circle Theorems
Alex
3 Topics
5 days

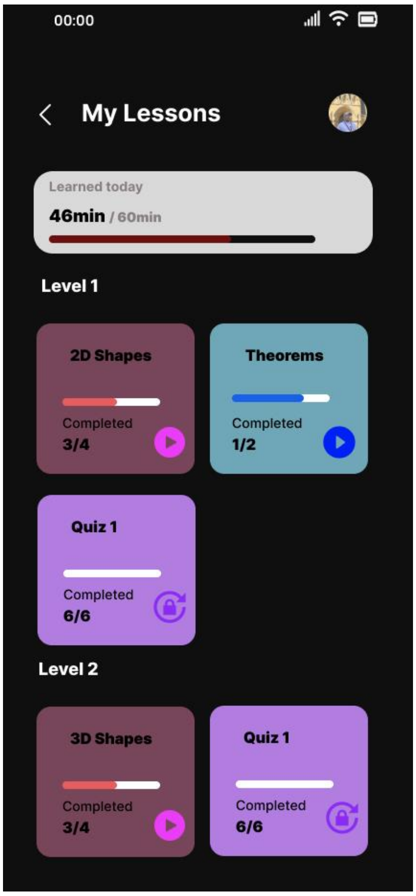

00:00
Traingles Theorem
Pythagoras' Theorem
Start
2 hour
Sine Rule
Start
30 min
Area of a Triangle
Start
4 hour

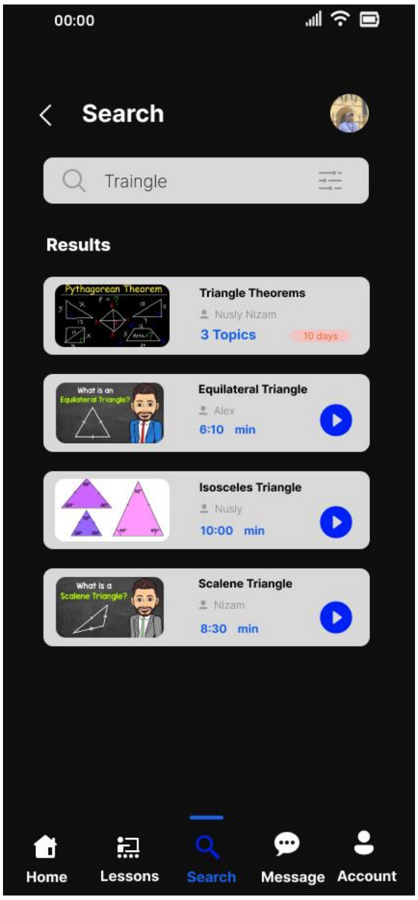

00:00
Circle Theorem
Angle at the Center
Start
2 hour
Tangent-Secant Theorem
Start
30 min
Inscribed Angle Theorem
Start
4 hour

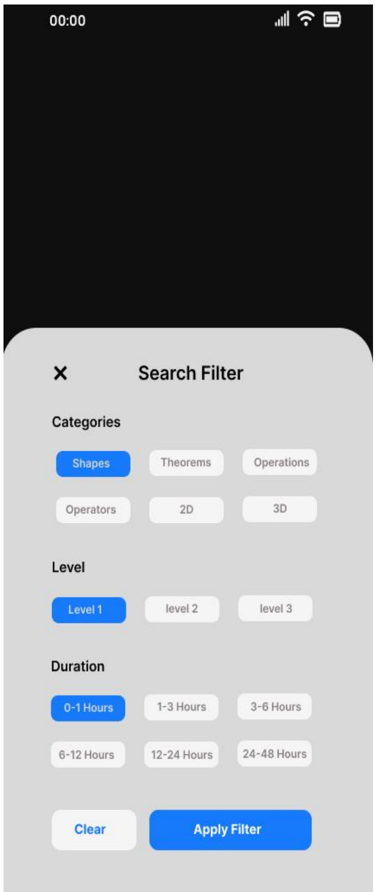

00:00
Operations
Learned today
Transformations
Nusly Nizam
4 Topics
10 days
Measurements
Alex
3 Topics
5 days

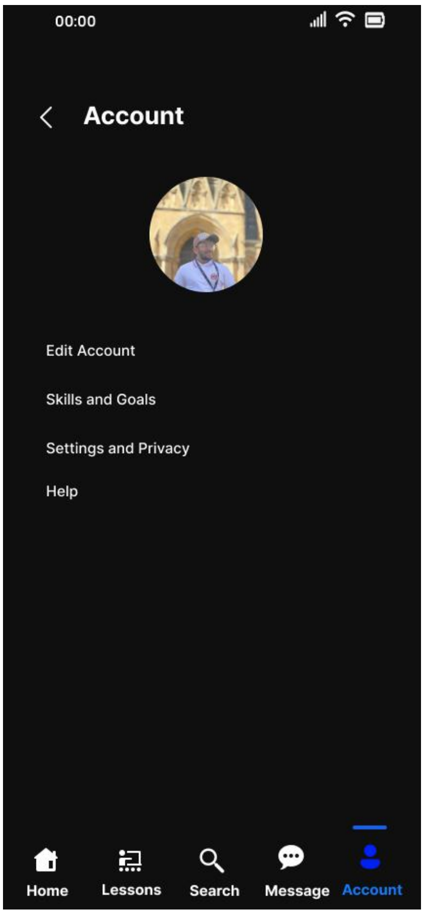

00:00
Operators
Learned today
Relational Operators
Nusly Nizam
3 Topics
3 hours
Arithmetic Operators
Alex
4 Topics
5 hours
Logical Operators
Alex
3 Topics
5 hours
Geometric Operators
Alex
3 Topics
3h 30min

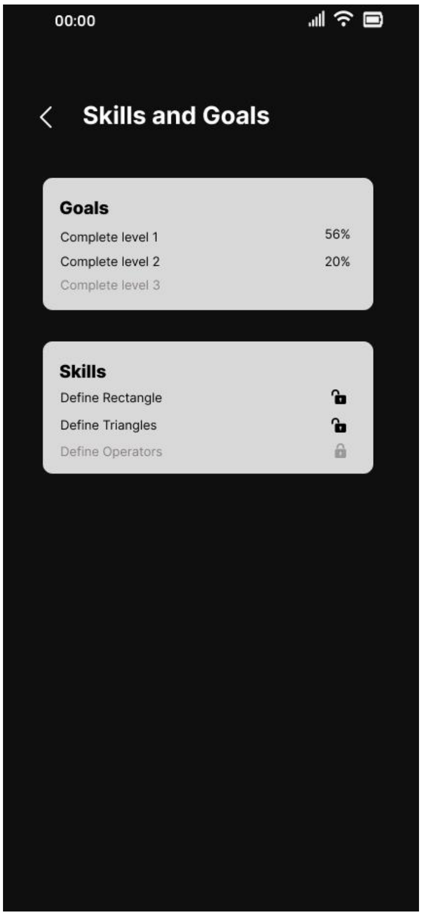

00:00
Data Types
Learned today
Primitive Types
Nusly Nizam
3 Topics
10 days
Reference Types
Alex
3 Topics
5 days

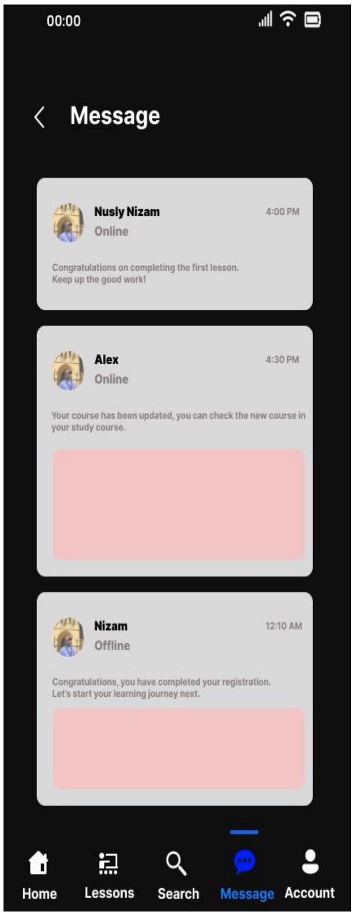

00:00
My Lessons
Learned today
46min / 60min
Level 1
2D Shapes
Completed
3/4
Theorems
Completed
1/2
Quiz 1
Completed
6/6
Level 2
3D Shapes
Completed
3/4
Quiz 1
Completed
6/6


00:00
Search
Traingle
Results
Triangle Theorems
Nusly Nizam
3 Topics
10 days
Equilateral Triangle
Alex
6:10 min
Isosceles Triangle
Nusly
10:00 min
Scalene Triangle
Nizam
8:30 min
Home
Lessons
Search
Message
Account


00:00
Search Filter
Categories
Shapes
Theorems
Operations
Operators
2D
3D
Level
Level 1
level 2
level 3
Duration
0-1 Hours
1-3 Hours
3-6 Hours
6-12 Hours
12-24 Hours
24-48 Hours
Clear
Apply Filter


00:00
Account
Edit Account
Skills and Goals
Settings and Privacy
Help
Home
Lessons
Search
Message
Account


00:00
Skills and Goals
Goals
Complete level 1
56%
Complete level 2
20%
Complete level 3
Skills
Define Rectangle
Define Triangles
Define Operators


00:00
Message
Nusly Nizam
4:00 PM
Online
Congratulations on completing the first lesson.
Keep up the good work!
Alex
4:30 PM
Online
Your course has been updated, you can check the new course in your study course.
Nizam
12:10 AM
Offline
Congratulations, you have completed your registration.
Let's start your learning journey next.
Home
Lessons
Search
Message
Account