Web Platform for Symbolic AI and Knowledge

Magdalena Maglizhanova[1,*,†], Veneta Tabakova-Komsalova[1,3,†], Veronica Dahl[2,*,†], Stanimir Stoyanov[1,3,†], Srebrin Srebrev[1,†] and Tanya Srebreva[1,3,†]

[1]*University of Plovdiv "Paisii Hilendarski", Plovdiv, Bulgaria*

[2]*Simon Fraser University, Burnaby, BC, Canada*

[3]*Centre of Excellence in Informatics and Information and Communication Technologies, Sofia, Bulgaria, Acad. G. Bonchev St., Block 2, 1113 - Sofia, Bulgaria*

[4]*Burgas State University "Prof. Dr. Assen Zlatarov", Burgas, Bulgaria*

Abstract

This paper presents a web-based platform for symbolic artificial intelligence, designed for both knowledge creators and end users. It has been developed in response to the growing interest in explainable and transparent AI, offering an alternative to the limitations of contemporary statistical approaches and large language models. Unlike these approaches, symbolic AI relies on the explicit representation of knowledge through facts and rules, combined with deterministic mechanisms for logical inference. The platform employs SWI-Prolog as its core inference engine, integrated with a server that manages user sessions and facilitates interaction with the logic layer. The architecture is multi-layered and modular, comprising a web interface, a business logic layer, and a data layer with domain-specific knowledge bases. The system supports multiple user roles—administrator, teacher, student, and knowledge user—allowing both the creation and the use of expert systems. From an educational perspective, the platform serves as an environment for learning artificial intelligence and logic programming through project-based and constructivist approaches. Students develop Prolog programs, while teachers validate and integrate the resulting knowledge bases. The platform encourages teamwork, international collaboration, and the development of analytical and logical thinking skills. The paper provides an overview of the IDEAS platform and its application in building artificial intelligence systems, where users are not just consumers, but active creators of intelligent solutions.

Keywords

web-based platform; knowledge representation; artificial intelligence education; symbolic artificial intelligence; logic programming; Prolog; explainable artificial intelligence (Explainable AI); project-based learning.

CEUR-WS.org/Vol-4311/PEG_regular_03.pdf

1. Introduction

Symbolic Artificial Intelligence emphasizes the explicit representation of knowledge, formal logic, and deterministic reasoning, where intelligent behavior is achieved through the manipulation of symbols and logical structures rather than through statistical learning from data. This approach is the basis for early research in the field of artificial intelligence and forms the basis for the development of expert systems, logic programming, and formal models of reasoning [1], [2]. In recent years, large language models (LLMs) have assumed a dominant role in practical AI applications, such as GPT, LLaMA, and Grok, which achieve impressive results through statistical learning on large sets of text data. However, several studies [3], [4] highlight their limitations, including a tendency toward hallucinations, lack of determinism, and difficulty in interpreting the decision-making process. These characteristics make them problematic for applications that require accuracy, traceability, and formal explainability, such as education, medicine, law, and expert advisory systems. In this context, there has been renewed interest in symbolic artificial intelligence and in hybrid approaches that combine symbolic and neural reasoning

PEG 2026

*Corresponding author.

†These authors contributed equally.

magdalenamaglijanova@uni-plovdiv.bg (M. Maglizhanova); v.komsalova@uni-plovdiv.bg (V. Tabakova-Komsalova); veronica_dahl@sfu.ca (V. Dahl); stani@uni-plovdiv.bg (S. Stoyanov); s.srebrev@uni-plovdiv.bg (S. Srebrev); tsrebreva@uni-plovdiv.bg (T. Srebreva)

0009-0000-4978-9282 (M. Maglizhanova); 0000-0002-0617-9844 (V. Tabakova-Komsalova); 0000-0002-3854-4260 (S. Stoyanov)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-03

[5], [6]. Symbolic methods offer transparency and control, while subsymbolic models contribute flexibility and efficiency in processing natural language [7]. The present work aligns with this trend by introducing a modern web-based platform—IDEAS, that implements classical symbolic AI through logic programming, without relying on probabilistic or statistical models. The main objective of this study is to support the introduction of logic programming, based on the Prolog language, into secondary education through the development and implementation of a modern web-based platform for symbolic artificial intelligence. The approach is based on the idea that AI education should be structured in a differentiated manner, taking into account the varying levels of prior knowledge, interests, and cognitive abilities of students.

This concept is consistent with our previous study [8], which argues for the need for a gradual and adaptive introduction of artificial intelligence content in secondary education. The authors emphasize that effective AI education requires a combination of conceptual understanding, practical activities, and opportunities for further development based on individual progress of the students. The differentiated approach enables both broad student participation and the creation of conditions for a more in-depth development of motivated and advanced learners. To achieve this goal, we propose a model characterized by three interrelated aspects:

1. **Mass and elite learning.** The mass component aims to introduce all students to the fundamentals of logic programming and symbolic models of knowledge representation. Students are trained to use existing logical knowledge bases and to formulate queries through which they can derive new knowledge by means of deduction. The elite component is intended for students with a deeper interest and a relevant background in mathematics or computer science. They not only use the system, but also develop their own expert modules, construct rules, and model knowledge. This two-level structure fully corresponds to the principles of differentiated instruction discussed in [8], where the importance of different levels of engagement—from user-level to creative-level—is emphasized.

2. **Teamwork.** The approach involves forming teams of both teachers and students. Teacher teams develop domain-specific knowledge bases and methodological scenarios, while student teams work on projects that involve the creation and testing of expert systems. In this way, AI education becomes a collective process of modeling, experimentation, and analysis, rather than passive adoption of ready-made technologies.

3. **Building a network of schools.** The third aspect involves the creation of a sustainable network of schools that share developed knowledge bases, instructional scenarios, and best practices. This enables the accumulation of expertise and the gradual expansion of the ecosystem of educational expert systems. This model of scalability and shared development is consistent with the principles of sustainable integration of AI in education embedded in the differentiated educational approach.

Through this model, logic programming is not presented as an isolated topic within secondary education but as an accessible and pedagogically structured domain that develops analytical, critical, and formal thinking already at the secondary school level. In this way, artificial intelligence becomes not merely a technology to be used but an environment to create, understand, and build intelligent systems.
The proposed platform employs SWI-Prolog as the primary mechanism for logical inference and decision-making. Prolog is a well-established declarative language for logic programming, widely used in expert systems and research applications, in which knowledge is represented through facts and rules, and conclusions are derived through formal logical mechanisms [9]. This approach enables deterministic behavior, reproducible results, and a clear relationship between input knowledge and derived conclusions.

A significant contribution of the system is **the dual role of the platform**. On the one hand, it functions as a tool for end users, who interact with expert knowledge through a web-based chat interface and receive logically grounded responses. On the other hand, the platform serves as an environment for creating artificial intelligence, in which users are not only consumers, but active

creators of intelligent systems. Students develop their own expert systems through logic programming, thus directly participating in the process of modeling knowledge and intelligent behavior. At the same time, the platform provides opportunities for teamwork. Teams can be formed among students within a single class, across different classes within the same school, between different schools, and at an international level. In addition, teams of teachers from different subject areas and different schools can be formed, encouraging interdisciplinary collaboration and the exchange of best practices. This paper presents the current state of the IDEAS platform.

From an educational perspective, this approach aligns with several studies [10], [11] that emphasize the importance of symbolic AI and logic programming for the development of analytical and critical thinking. Instead of working with "black boxes," learners understand how knowledge is structured, how rules lead to inferences, and how logical errors result in incorrect results. This transforms artificial intelligence from an abstract technology into a comprehensible and verifiable tool. In a broader scientific context, the presented system demonstrates how classical ideas of logic programming can be successfully adapted to modern web and cloud environments. Through a clear separation between the knowledge base and the inference engine, as well as a modular architecture, the platform adheres to established principles of knowledge-based systems [12], while also offering the accessibility and scalability characteristic of modern web applications. In this way, the development represents a practical implementation of symbolic artificial intelligence in the context of 2025–2026 technologies and lays the foundation for future neuro-symbolic extensions.

2. System architecture

The architecture of the proposed platform follows a modular and multi-layered design (Fig. 1), combining the classical principles of knowledge-based systems with modern web and cloud technologies. This approach aims to ensure a clear separation between the user interface, request management, and logical reasoning, thereby guaranteeing security, scalability, reproducibility of inferences, and ease of long-term maintenance. The system is logically divided into server and client components. The server component implements the core functionality, data processing, and knowledge management, while the client component provides interaction with end users through a graphical interface.

Multi-layer Architecture. The architecture follows the classical three-layer model, which includes a presentation layer, a business logic layer, and a data layer. Each of these layers performs well-defined functions and communicates with adjacent layers through clearly defined interfaces.

Data Layer. The data layer is responsible for the storage and management of informational resources. Its structure includes multiple domain-specific knowledge bases, organized by content areas. These knowledge bases can store both structured data, such as information about users, roles, and courses, and unstructured data, including textual materials, documents, and educational resources. The primary function of this layer is to ensure reliable storage, fast access, and data consistency. Access to the knowledge bases is provided through an abstract interface, which isolates the business logic from the underlying database management technology. This allows for replacement or expansion of the storage system without significant changes to other components.

Business Logic Layer. The business logic layer represents the central control mechanism of the system. It processes all incoming requests, applies defined rules, and coordinates access to the knowledge bases. This layer implements the core functionality of the platform, including intelligent request processing. When a request is received from a user, input data is validated and access rights are verified. The request is then analyzed, and the appropriate processing mechanism is determined. If information retrieval is required, the system selects the relevant domain-specific knowledge base and performs the search. If an AI component is involved, semantic analysis and generation of a personalized response are carried out. Within this layer, a user management module also operates, handling registration, authentication, authorization, and role management. Additionally, a mechanism is maintained for tracking learner activity and progress. Coordination between the individual functional components is managed through internal process-control logic, ensuring the correct sequence of operations.



Figure 1: Platform Architecture

Presentation Layer. The presentation layer provides the visualization of the system and the interface for user interaction. It is implemented as a web application, accessible through a standard internet browser. The interface allows users to input queries, view results, and manage their profiles according to their roles.

Communication between the presentation layer and the business logic layer is carried out through a secure application programming interface. Data exchange is performed using standardized formats, ensuring compatibility and easy integration with external systems. In addition to the multi-layered structure, the system is built on a modular principle. Each module implements a specific functionality and can be developed, tested, and maintained independently of the others. This approach reduces dependencies between components and facilitates system expansion. The knowledge base module is responsible for organizing and managing domain-specific content. The intelligent processing module analyzes incoming queries and generates adaptive responses. The user management module controls access and roles within the system, while the interface module provides the visual environment for interaction. The communication module handles data exchange between the client and server components using standardized protocols. The modular design allows for the addition of new functionalities, such as a new domain-specific knowledge base or an enhanced AI algorithm, without disrupting the overall system architecture.

3. Platform for AI Creators

One of the main conceptual differences between the proposed platfom and most contemporary AI-based systems is its focus on creation rather than solely on the use of artificial intelligence. While widely adopted AI solutions treat the user as a passive consumer of automatically generated results, the current platform is designed as a knowledge engineering environment, in which users actively build, structure, and validate intelligent systems. This approach aligns with contemporary research [13, 14], emphasizing the importance of human-in-the-loop and human-centered AI for reliable and explainable intelligent systems.

At the core of the platform is the understanding that artificial intelligence can and should be consciously constructed through the formalization of knowledge, rather than merely "trained" using statistical methods. AI creators within the platform—students, teachers, or domain experts—develop intelligent behavior by defining facts and logical rules in Prolog. This approach reflects the classical principles of symbolic AI but places them in a modern web context, lowering the access barrier and broadening the pool of potential intelligent system creators. Contemporary studies [15] indicate that symbolic AI plays a key role in building systems that are more transparent, easier to verify, and more human-controllable compared to purely neural models. In this sense, the platform not only implements symbolic AI technologically but also follows current recommendations for constructing reliable AI.

The platform enforces a clearly defined role model that reflects both technical and pedagogical requirements. Three primary user roles are distinguished: administrator, teacher/instructor, and student. This separation follows the principles of role-based access control and least-privilege policies, widely applied in modern educational and intelligent systems.

The administrator is responsible for the overall management of the system, including access control, approval of teachers, and maintenance of thematic domains.

The teacher acts as a knowledge engineer and curator—creating educational domains, reviewing and validating knowledge bases developed by students, and integrating them into the shared repository.

The student/creator actively participates in the knowledge creation process by developing their own programs, experimenting with logical rules, and analyzing the outcomes of inference processes.

Such a model of distributed responsibility aligns with contemporary research on collaborative knowledge building and AI-supported education [16, 17]. It demonstrates that active involvement of learners in knowledge creation leads to deeper understanding and higher engagement. From a pedagogical perspective, the platform follows a project-based and constructivist learning model, in which students learn by creating real artifacts—in this case, Prolog programs. This approach is supported by recent studies [18], which indicate that learning through the creation of symbolic models and formal knowledge representations fosters logical thinking, metacognitive skills, and understanding of cause-and-effect relationships. Unlike using prebuilt AI tools, where the underlying logic remains hidden, here learners can clearly trace how changes in facts or rules lead to different inferences. This transforms the inference process into a didactic tool and allows students to develop an intuition for how an intelligent agent "thinks"—a key aspect of AI education [19].

Knowledge management in the platform follows the principles of modern knowledge engineering, treating knowledge as an independent, versionable, and shareable resource. Knowledge bases are organized as Prolog files, grouped by thematic domains, and stored in a cloud environment. This enables dynamic addition, updating, and reuse of knowledge without changes to the logical core. Contemporary research [20] emphasizes that effective knowledge management is critical for scalable symbolic and neuro-symbolic systems, especially when multiple human creators are involved. The platform implements these principles through a clear separation between the inference engine and domain-specific knowledge, as well as through controlled processes for approval and integration of new knowledge bases.

The inference process in the system is carried out entirely through deterministic logical reasoning. Once a domain is selected, the corresponding knowledge is loaded into an isolated working environment, after which user queries are processed by the Prolog inference engine. Each response is a direct consequence of the available facts and rules, allowing full traceability and explainability of results [21, 22]. This approach aligns fully with modern requirements for Explainable AI (XAI), according to which intelligent systems should not only provide answers but also allow understanding of the logic behind them [23, 24]. In an educational context, this is particularly important, as explainable inference supports the development of logical thinking, reflective reasoning, and learners' ability to analyze cause-and-effect relationships within the formal structures of knowledge.

4. Sample scenario for using the platform for AI creators

The platform is used in an educational setting for computer science and natural sciences, with the aim of having students develop Prolog programs while the teacher supervises and evaluates the knowledge. For broader functionality and use by international participant groups, the platform is available in three languages (Bulgarian, Spanish, and English). Here we present a real scenario executed by all system roles.

The administrator initially configures the system by creating thematic domains (e.g., biology, chemistry, animals, mountains, and others). They approve teacher registrations and monitor the activity and stability of the platform. After this initial setup, the administrator does not participate directly in the teaching process.

Pedagogical context and role of the teacher. The teacher plays a leading role in planning, organizing, and evaluating educational materials. They work collaboratively with students to create Prolog programs, guiding and supporting them throughout the process. The work is organized in five main stages, in each of which the teacher actively participates to ensure high-quality instruction, correctness of knowledge, and timely substantive feedback. Within the system, the teacher performs the following key functions: organizer of the educational process, community moderator, creator of learning content and challenges, project evaluator, and guarantor of the quality of the knowledge base. After the initial administrative configuration carried out by the administrator (defining domains, approving teachers, and monitoring system stability), the teacher assumes full control over the educational activity within their communities. The teacher registers using an email address. After approval by the administrator, they gain access to a personalized Teacher Dashboard (Fig. 2).

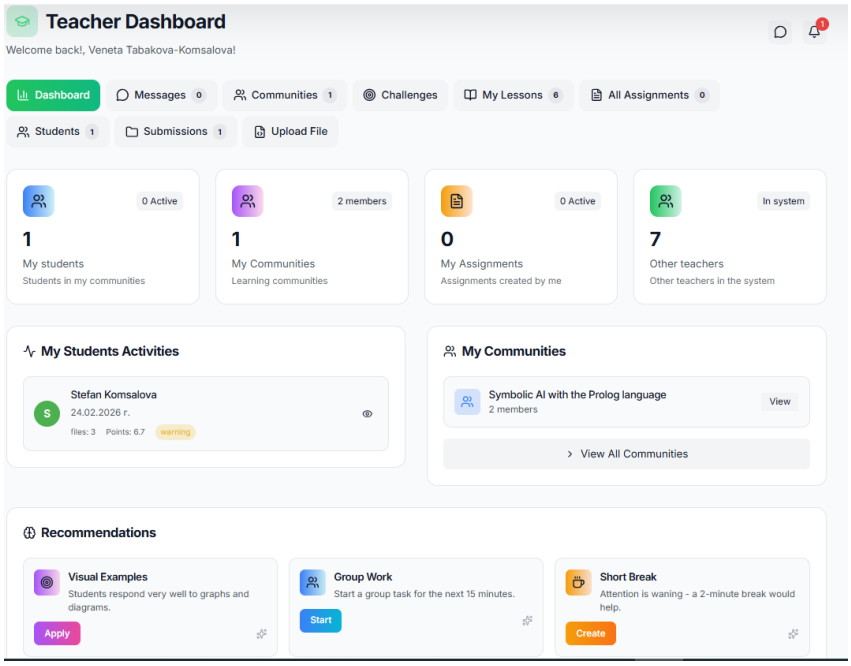


Figure 2: Teacher Dashboard

The management dashboard displays created communities; the number of students in each community; active assignments and challenges; new join requests; recently submitted projects; activity statistics; and an internal messaging system. This central panel allows effective monitoring of the educational process. Here, the teacher creates a learning community by defining its name, subject, description, educational level, and a unique join code. Students apply to join by entering the provided code. Requests appear on the teacher's dashboard and require approval. Once confirmed, students become active participants in the respective community. This mechanism ensures a controlled learning environment and a clear group structure.

Designing the Educational Model The educational model begins with building a comprehensive framework in which the teacher defines key parameters of the activity. They decide how students will be organized — individually, in pairs, or in small teams — and formulate the specific learning objective, such as mastering logical programming, modeling knowledge in a particular subject, or developing analytical thinking. The teacher also selects the subject and conceptual framework, which may be biology, chemistry, physics, or another suitable domain for developing a logical program, for example, disease diagnosis, chemical reactions, physical laws, or organism classification.

This step ensures that the task aligns both with the curriculum and the students' age and preparation level.

Preparation of Educational Materials. The teacher creates lessons containing: title; description; objectives; expected outcomes; key concepts; and sample logical constructions.

Example Learning Module: Creating an Expert System for the Rhodope Mountains in Bulgaria. The modeling target is knowledge related to the Rhodope Mountains. Students research reliable sources, identify relevant objects, determine their attributes, formalize the knowledge as facts and rules in Prolog, and create a logical inference mechanism. The goal is not only to generate code but to build a fully formalized expert system.

Creating Assignments. An assignment is a structured learning activity with clearly defined criteria. When creating it, the teacher formulates detailed requirements for developing an expert system, specifying a clear topic and scope—for example, a system for identifying animals based on characteristics or diagnosing chemical reactions based on symptoms. The teacher selects reliable and verified sources from which information can be extracted, structured, and converted into code in the form of knowledge. Requirements for the system's structure are also clarified, including the use of facts, rules, the main predicate start/0, and a user menu for interaction.

Additionally, evaluation criteria are defined, covering the correctness and completeness of facts and rules, logical consistency and absence of errors, system functionality, code quality in terms of readability, comments, and modularity, as well as the degree of originality and creativity. All this information is provided to students in advance, so they have a clear understanding of what is expected of them.

Organizing Challenges. Challenges serve as a competitive mechanism for knowledge assessment. They may include solving a logical case, decoding a logically formulated message, constructing the most efficient rule, or working under a time constraint. This tool stimulates analytical thinking, deduction, and formal structuring of knowledge.

Tracking the Learning Process. The platform provides the ability to monitor student activity by tracking deadlines, reviewing submitted assignments and their versions. It also allows progress control. This monitoring supports timely pedagogical intervention.

Evaluation and Grading. After a project (*.pl file) is submitted, it is stored in the cloud and becomes accessible in the teacher's review panel. The teacher can execute logical queries through the integrated chat interface. Evaluation is based on the correctness of facts, logical consistency, absence of contradictions, determinism of inferences, and explainability of conclusions. The teacher applies pre-defined criteria to ensure objective grading, tests the code in the system via /prolog/command or by direct loading, and assigns a quantitative grade accompanied by detailed feedback. The feedback highlights the strengths of the work, points out any logical errors or inaccuracies in facts, and provides guidance for improvement. When a project is approved, the teacher integrates high-quality student work into the corresponding thematic knowledge base by uploading the approved code to the Supabase system, making it part of the collective repository and available for other students or future lessons.

This cycle allows the teacher not only to evaluate but also to purposefully enrich the platform with verified, educationally valuable knowledge created by students. In this way, the process promotes active learning through creation, critical assessment, and knowledge sharing in a controlled environment based on symbolic artificial intelligence.

Integration into the Knowledge Base. Approved projects are integrated into the system's collective knowledge base. This makes the knowledge accessible to other learners, builds an expandable expert repository, ensures accumulation of verified content, and guarantees the sustainability of the learning process.

Through the messaging system, the teacher can send messages to all teachers on the platform or only to students within their own community. Using the system allows the teacher to manage the learning process in a structured way, formalize knowledge, ensure transparency of logical inferences, maintain objectivity in assessment, and build a stable, verified knowledge base. The platform transforms the traditional learning model from information reproduction to the construction of formal expert systems, with the teacher acting as a methodologist, moderator, and guarantor of the logical validity of knowledge.

Using the System as a Student/Creator. Within the platform, the student acts as a knowledge creator, with the primary task of modeling a subject domain in the form of Prolog programs. The learner is not a passive consumer of information but an active constructor of formalized knowledge, which can later be integrated into the system's collective knowledge base.

Registration and Joining a Community. After registration, the student gains access to the Student Dashboard (Fig. 3), which includes: a list of communities; active assignments; ongoing challenges; tasks for the day; messaging system; and work recommendations.

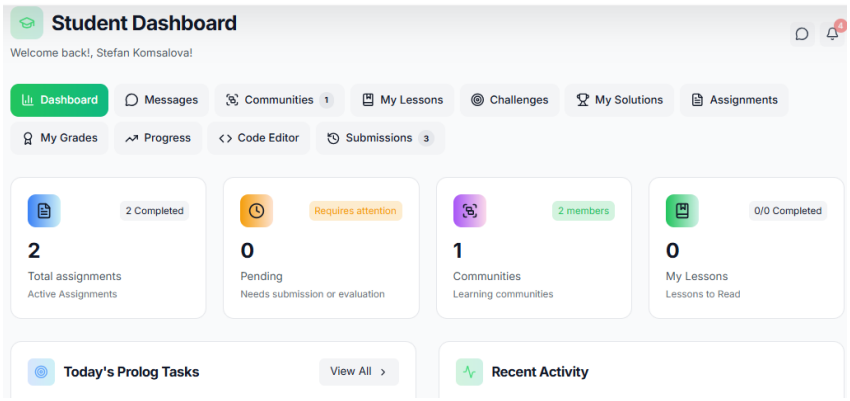


Figure 3: Student Dashboard

To participate in a learning activity, the student enters the code of the corresponding community, provided by the teacher, and submits a request to join, then waits for approval. Once approved, the community becomes active in their profile.

Within the community, the student has access to: lesson description; learning objectives; instructions; minimum requirements; deadline; and a sample Prolog file template.

Working on an Assignment. When opening an assignment (Fig. 4), the system automatically loads the Prolog file template provided by the teacher and creates an editing workspace. In this environment, the student can directly add facts and define rules, thereby creating logical dependencies between objects. The goal is to build an internally consistent logical system.



Figure 4: Assignment template

Students, individually or in teams, gather and analyze scientific information from reliable sources, transform it into facts and logical rules in Prolog, and build the system's structure, including the main predicate start/0 and an interactive menu for user queries.

During the development of their systems, they must ensure that the code runs directly without errors, functions correctly when tested in the SWI-Prolog online environment, and meets the requirements specified in the assignment.

During the development process, the teacher can provide guidance, examples, and consultations without offering ready-made solutions.

Submitting the Project. Upon completing their work, the student uploads the *.pl file to the assignment via the interface. The file is stored in a cloud repository, and the project automatically becomes available for review by the teacher. The system confirms successful submission and records compliance with the deadline.

Participation in Challenges. Challenges represent a competitive component of learning. The student can solve logical cases, decode formalized messages, create the most efficient rule, and work under time constraints. This mechanism stimulates rapid deduction, analytical thinking, and knowledge structuring under pressure.

Receiving Feedback. After evaluation, the student receives the numerical grade assigned by the teacher, written feedback (Fig. 5), and, if needed, specific recommendations and instructions for improvement. This process supports the development of self-monitoring skills, logical precision, formal reasoning, and structured knowledge organization.

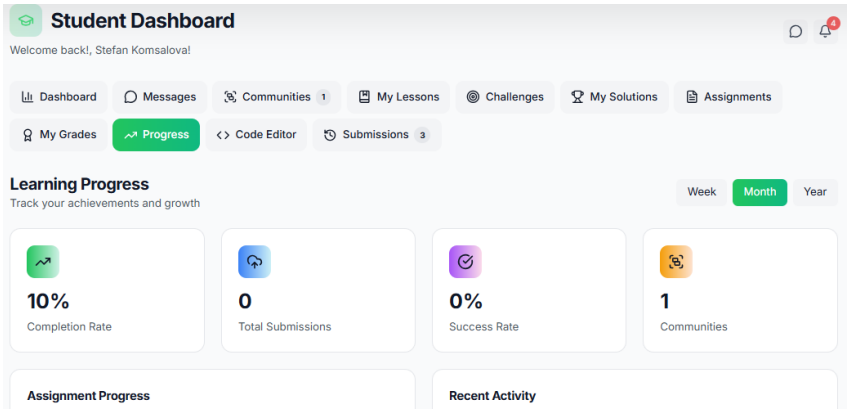


Figure 5: Feedback on a submitted assignment

Integration into the Shared Knowledge Base. Once approved by the teacher, the project can be integrated into the shared knowledge base, making it available for future lessons and accessible to other students through the chat interface. In this way, the student contributes to expanding the knowledge base and gains a tangible contribution to a developing logical knowledge system.

Using the system as a student/creator leads to active knowledge construction and the development of formal logic skills. In this way, analytical and structured thinking is fostered. In this model, the student transitions from being a consumer of information to a creator of formalized, verifiable, and explainable knowledge, which represents a core objective of education in symbolic artificial intelligence.

5. Conclusion and Future Development.

This work presents a cloud-based platform for symbolic artificial intelligence that combines the principles of logical programming with modern web infrastructure. Unlike dominant LLM-based solutions, it emphasizes explicit knowledge representation and deterministic logical reasoning, making them accessible to a wide range of AI users and creators. By using Prolog as the core inference engine, the system ensures transparency, repeatability, and explainability of conclusions—essential characteristics

for education and applications requiring trust and control. The architecture demonstrates that symbolic AI can be successfully integrated into a modern cloud environment. A significant contribution is the focus on humans as AI creators. Through clearly defined roles and an educational model based on building expert systems, the platform encourages active participation in knowledge engineering and supports learning in artificial intelligence, logical thinking, and formal knowledge modeling.

Future research could evaluate the educational impact of using symbolic AI and the cognitive benefits of learners' participation in creating logical programs.

In conclusion, the platform demonstrates that symbolic artificial intelligence remains strategically important in the era of generative models, offering an explainable and sustainable alternative that complements LLM approaches and lays the foundation for future hybrid intelligent systems.

Acknowledgments

The work was partially supported by Center for Excellence in Informatics and ICT under grant No. BG16RFPR002-1.014-0018-C01, funded by the Research, Innovation and Digitalization for Smart Transformation Program 2021–2027 and co-funded by the European Union.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] A. Newell, H. A. Simon, Computer science as empirical inquiry: Symbols and search, Communications of the ACM 19 (1976) 113–126. doi:10.1145/360018.360022.

[2] S. J. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, 4 ed., Pearson, 2021.

[3] E. M. Bender, T. Gebru, A. McMillan-Major, S. Shmitchell, On the dangers of stochastic parrots, in: Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (FAccT), 2021, pp. 610–623. doi:10.1145/3442188.3445922.

[4] G. Marcus, The next decade in ai, arXiv preprint arXiv:2002.06177 (2020).

[5] Y. J. Phua, K. Inoue, Variable assignment invariant neural networks for learning logic programs, in: Neural-Symbolic Learning and Reasoning, Springer, 2024. doi:10.1007/978-3-031-71167-1_3.

[6] D. Cunnington, M. Law, J. Lobo, A. Russo, The role of foundation models in neuro-symbolic learning and reasoning, in: Neural-Symbolic Learning and Reasoning, Springer, 2024. doi:10.1007/978-3-031-71167-1_5.

[7] F. Manigrasso, S. Schouten, L. Morra, P. Bloem, Probing llms for logical reasoning, in: Neural-Symbolic Learning and Reasoning, Springer, 2024. doi:10.1007/978-3-031-71167-1_14.

[8] V. Tabakova-Komsalova, M. Maglizhanova, S. Stoyanov, A differentiated approach to learning ai in secondary school, Education and Technologies 16 (2025) 206–212.

[9] W. F. Clocksin, C. S. Mellish, Programming in Prolog, 5 ed., Springer, 2003. doi:10.1007/978-3-642-55481-0.

[10] J. Minker, Logic-Based Artificial Intelligence, Kluwer Academic, 2000.

[11] R. Kowalski, Computational Logic and Human Thinking, Cambridge University Press, 2011. doi:10.1017/CBO9780511984747.

[12] GeeksforGeeks, Expert systems in ai, https://www.geeksforgeeks.org/artificial-intelligence/expert-systems/, 2025. Accessed: 2026-03-10.

[13] B. Shneiderman, Human-centered artificial intelligence, International Journal of Human–Computer Interaction 36 (2020) 495–504. doi:10.1080/10447318.2020.1741118.

[14] S. Amershi, et al., Guidelines for human-ai interaction, in: Proceedings of the CHI Conference on Human Factors in Computing Systems, 2019. doi:10.1145/3290605.3300233.

[15] G. Marcus, E. Davis, Rebooting AI, ACM, 2019.
[16] W. Holmes, et al., Artificial Intelligence in Education, Center for Curriculum Redesign, 2019.
[17] H. Khosravi, et al., Generative ai and learning analytics, Journal of Learning Analytics 12 (2025) 1–11. doi:10.18608/jla.2025.8961.
[18] R. Lister, et al., Further evidence of a relationship between explaining, tracing and writing skills, SIGCSE Bulletin 41 (2009) 161–165. doi:10.1145/1595496.1562930.
[19] D. Touretzky, et al., Envisioning ai for k-12, in: Proceedings of the AAAI Conference, 2019. doi:10.1609/aaai.v33i01.33019795.
[20] S. C. Shapiro, Knowledge representation, Computational Linguistics 27 (2001).
[21] R. Kowalski, Logic for Problem Solving, North-Holland, 1979.
[22] L. Sterling, E. Shapiro, The Art of Prolog, MIT Press, 1994.
[23] T. Miller, Explanation in artificial intelligence, Artificial Intelligence 267 (2019) 1–38. doi:10.1016/j.artint.2018.07.007.
[24] R. Guidotti, et al., A survey of methods for explaining black box models, ACM Computing Surveys 51 (2018). doi:10.1145/3236009.