MounouZinc:
A MiniZinc-Based University Timetabling Tool Integrating AI Optimization and Web Automation

Tony Rouhayem[1], Youssef Bakouny[1], Laurent Drapeau[2], and Melhem El Helou[1]

[1] Faculty of Engineering and Architecture, Saint Joseph University of Beirut, Lebanon
[2] CNRS, IRD, INRAE, CEREGE, Aix-en-Provence, Aix Marseille University, Marseille, France

Abstract. University course timetabling is an NP-hard optimization problem defined by complex combinations of academic, pedagogical, and resource-related constraints. While recent advances in constraint programming and hybrid optimization have improved model expressiveness, institutions still lack integrated tools that connect optimization, data collection, recommendation, and student-support capabilities within a unified platform. This paper presents MounouZinc, a tool for university timetabling, designed as a modular built on MiniZinc. It integrates a PostgreSQL backend, a Python automation layer, and a web-based interface for collecting professor availability and timeslot preferences. The system integrates an AI-driven professor–course matching agent, which prioritizes professor assignments based on specialization and course descriptions, and a conversational chatbot that supports both new applicants (FAQ services) and current students (iAdvisor-style course selection and enrollment assistance). Experimental evaluations were conducted on synthetically generated datasets designed to reflect real institutional scenarios demonstrate that MounouZinc produces feasible, high-quality timetables that align with diverse academic policies while providing end-to-end decision-support functionalities rarely found in existing UCTP systems. MounouZinc advances beyond basic solving to offer an integrated decision-support platform.

Keywords: MiniZinc · Timetabling · Scheduling · Constraint Programming · Optimization · Artificial Intelligence · Chatbot · iAdvisor

1 Introduction

The University Course Timetabling Problem (UCTP) is a well-known NP-hard combinatorial optimization problem involving the assignment of courses to timeslots, rooms, and professors under various institutional and resource constraints

[1]. Its complexity, coupled with evolving academic requirements, has driven extensive research across exact, heuristic, and hybrid approaches [2][3]. Traditional manual scheduling methods are increasingly inadequate for handling modern timetabling demands, including fairness, flexibility, and scalability.

Constraint Programming (CP) and mixed-integer programming provide expressive frameworks for modeling complex UCTP variants [4][5]. Meanwhile, advancements in educational AI have introduced intelligent recommendation systems, explainable planning tools, and conversational agents for academic support [7][8][9]. However, these solutions are typically fragmented and lack integration into a unified system.

While constraint programming and mathematical optimization provide expressive modeling capabilities, and recent advances in educational AI enable intelligent recommendation and conversational support, existing solutions remain largely fragmented. In practice, institutions require integrated systems that combine optimization, data acquisition, and decision-support functionalities within a unified workflow.

To address this gap, we present MounouZinc, an extensible timetabling and academic decision-support system that tightly integrates constraint-based optimization, AI-driven recommendation, and conversational guidance within a unified architecture. Built on MiniZinc, the system supports flexible constraint modeling, modular solver configurations, and seamless integration with a PostgreSQL backend. It features a web-based interface for capturing professor availability, an AI-driven professor–course matching module leveraging semantic similarity between faculty expertise and course descriptions, and a conversational chatbot for academic advising and FAQ support.

MounouZinc provides a unified framework that bridges optimization techniques with institutional workflows and AI-driven services, serving both as a research platform and a deployable solution for real-world academic environments. The proposed framework adopts a modular MiniZinc-based modeling approach that enables dynamic activation of constraints and solver interchangeability, facilitating adaptability to diverse institutional requirements. Furthermore, MounouZinc implements an end-to-end automated workflow encompassing data acquisition, preference modeling, optimization, validation, and deployment. Experimental evaluation demonstrates the system's feasibility and scalability, along with measurable improvements in solution quality compared to baseline approaches.

2 Related Work

Research on the University Course Timetabling Problem (UCTP) has a long history, with surveys and case studies highlighting its combinatorial complexity and diverse institutional requirements. Early surveys such as [3] synthesize exact, heuristic, and metaheuristic approaches, while more recent work [2] provides updated taxonomies of models, objectives, and datasets, emphasizing trends such as robustness, fairness, and hybrid optimization methods. Educational timetabling research spans exact, heuristic, and hybrid methods, with recent surveys consolidating formulations and benchmarks across curriculum-, course-, and resource-based variants. In particular, [10] unifies problem definitions and benchmark suites, reinforcing the importance of standard testbeds for reproducible evaluation.

2.1 Real-World UCTP formulation

Recent works focus on rich, real-world UCTP formulations aligned with practical requirements. For instance, [4] proposes a multi-objective mixed-integer programming model balancing institutional criteria, while [11] addresses accreditation-constrained timetabling using MIP and heuristics. Robust timetabling under disruptions is studied in [12], and [5] presents a detailed case study using modern constraint- and mathematical-programming techniques, emphasizing transparency and maintainability.

2.2 Flexible and automated timetabling systems

There is growing interest in flexible and automated timetabling systems. A system in [6] combines mathematical modeling with an implementation-oriented architecture to support deployment in institutional environments. Similarly, [13] addresses business-school timetabling with multi-section courses, room stability, and professor preferences, highlighting the need for unified models integrating institutional policies and constraints. These align with MounouZinc's focus on modular design and system integration.

2.3 AI-driven course recommendation and academic planning

Another research strand focuses on AI-driven course recommendation and academic planning. Studies show that recommender systems can support course selection using performance, interests, and program requirements [7]. Broader perspectives on AI-based recommendation are discussed in [14], while [15] explores machine- and deep-learning models for academic planning. Recent work [8] emphasizes explainable and LLM-based approaches. Unlike these student-focused systems, MounouZinc extends recommendation to professor–course matching based on specialization and course characteristics.

2.4 Chatbots and conversational agents in higher education

A third line of research concerns chatbots in higher education. Reviews such as [9] highlight their growing role in student support and administrative assistance. Empirical studies [16] report benefits in scalability and accessibility, alongside challenges in data quality. Work such as [17] shows chatbots are effective for routine advising tasks, and case studies [18] confirm their utility for FAQs and student guidance. Unlike standalone chatbot systems, MounouZinc integrates conversational support directly into timetabling and course management workflows.

2.5 Research gap

Existing literature provides strong but largely fragmented contributions across timetabling, AI-based recommendation, and academic chatbot systems; however, no unified platform simultaneously integrates MiniZinc-based timetabling, AI-driven professor–course matching, structured preference collection, and conversational support. MounouZinc addresses this gap by delivering an end-to-end framework that combines optimization, recommendation, and interactive guidance within a single architecture, unifying paradigms that prior works treat in isolation—whether optimization-focused [4][5], recommendation-driven [7][8], or deployment-oriented systems lacking embedded intelligence and conversational capabilities [6][11]—thereby enabling both automated scheduling and enhanced, data-driven decision-making.

3 System Overview

This section provides an overview of the system's design principles, main system components, and typical workflow, illustrating how the tool supports the entire life-cycle of university course timetabling shown in Figure 1 from initial setup, data acquisition and preference collection, modeling, optimization, validation, timetable dissemination, and the intelligent student-facing iAdvisor module.

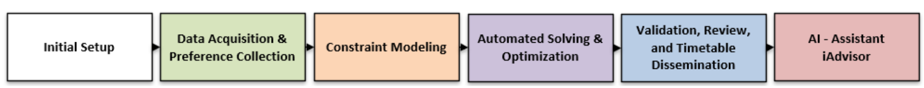

Fig. 1. Life Cycle

3.1 Design Principles

The development of MounouZinc was guided by four key goals that reflect the practical challenges faced by academic institutions:

1. **Modularity and Extensibility** Institutions differ widely in their organizational structures and scheduling policies. To accommodate this diversity, MounouZinc separates the data layer, constraint modeling layer, solver layer, and web interface. This modularity allows administrators to select the hard constraints to apply, incorporate the institutional rules by adjusting the weight of soft constraints, or switch solvers without modifying the core components.
2. **Usability for Non-Technical Staff** The scheduling is often carried out by administrative personnel rather than optimization experts. MounouZinc provides intuitive web interfaces for editing course data, capturing professor availability, visualizing schedules, and identifying constraints. The system is

designed so that most tasks require no direct interaction with MiniZinc or solver code. Everything can be handled through the web application.

3. **Data-Driven and Institution-Integrated Operation** Real deployments require synchronization with student information systems, HR data, and course catalogs. MounouZinc emphasizes structured data import and supports integration via APIs, CSV uploads, or database connectors.
4. **Decision Support** Beyond generating feasible schedules, MounouZinc offers an AI-based professor–course matching assistant. This assists planners in making informed adjustments and improves transparency for stakeholders. In addition, the AI chatbot will help students better select the courses in which to enroll.

3.2 Main System Components

As shown in Figure 2, MounouZinc is organized into five primary components that together support a complete end-to-end university timetabling workflow. These components define the functional structure of the system and establish the foundation upon which the architectural and implementation details in section 4 are built.

1. **Data and Persistence Layer (Postgres-Based)** This layer provides authoritative storage for all institutional data required during timetable generation. A normalized PostgreSQL schema maintains information on courses, professors, rooms, academic plans, institutional policies, and declared preferences. The schema ensures referential integrity, minimizes redundancy, and facilitates integration with Student Information Systems (SIS), HR databases, and curriculum management platforms. This layer serves as the central repository upon which all modeling and optimization tasks depend.
2. **Modeling and Solver Layer (MiniZinc-Based):** Institutional data is translated into MiniZinc models that encode both feasibility rules (hard constraints) and institutional preferences (soft constraints). The modeling layer is composed of modular constraint components covering professor availability, room compatibility, workload rules, accreditation requirements, and pedagogical structures. MiniZinc's solver-independent architecture enables MounouZinc to operate with multiple backends—including Chuffed, CP-SAT, and LCG-based solvers—allowing experimentation and solver substitution without changes to the underlying model.
3. **Web Interface and Interaction Layer (Node.js-Based):** The browser-based interface of the platform, powered by a Node.js and Express backend, enables administrators and professors to interact with the system intuitively and efficiently. Administrators can configure constraints, visualize timetables by room, professor, course, or study program, inspect violations, and manage scenario metadata. Professors use dedicated tools to declare availability, submit preferences, and review assigned teaching loads. Designed for non-technical users, this layer abstracts all modeling and solver complexity behind interactive workflows.

4. **AI-Assisted Services Layer:** MounouZinc incorporates intelligent components that extend its capabilities beyond traditional constraint solving by supporting decision-making, reducing administrative effort, and enhancing transparency for multiple stakeholders. These AI-driven features include preference ingestion mechanisms that transform availability and user inputs into structured model constraints, an NLP-based professor–course matching assistant that leverages teaching history, specialization data, and course descriptions to produce ranked assignment recommendations, and a Retrieval-Augmented Generation (RAG)-based conversational assistant that serves both as a FAQ and admissions tool for prospective students ("Chatbot") and as an academic advising agent (“iAdvisor”) for current students, offering guidance on course selection, prerequisites, and enrollment constraints.
5. **Automation and Orchestration Layer (Python-Based):** A Python middleware coordinates data handling, model construction, solver invocation, and result processing. It generates MiniZinc instances from database records, assembles relevant constraint modules, executes solvers with configured strategies, parses solution outputs, and injects final schedules back into the database. This layer exposes APIs to the web interface and maintains detailed logs to ensure reproducibility, traceability, and operational reliability across scheduling cycles.

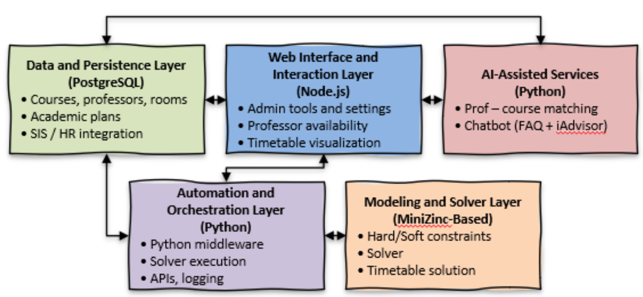

Fig. 2. High Level System Architecture

3.3 Typical Workflow

MounouZinc supports a structured and repeatable scheduling workflow aligned with real institutional practices. A typical cycle proceeds as follows:

1. **Data Import and Synchronization:** Administrators import course lists, rooms, professor profiles, timeslots, and academic plans from institutional databases or spreadsheets.

2. **Preference and Availability Collection:** Professors specify preferred teaching days and times. With the help of an AI-based tool, academic coordinators specify course-professor, and timeslot/room preferences.
3. **Constraint Configuration:** Administrators adjust institutional policies, enable optional constraint modules, assign penalty weights to soft constraints, and verify academic plan requirements or accreditation conditions.
4. **Model Generation and Solving:** The orchestration layer compiles all data and constraints into MiniZinc models. Solvers are executed under specified time limits or optimization goals. Final results are stored in the database.
5. **Visualization and Diagnostics:** Generated timetables along with statistical charts are visualized through the Web application. Users can access their own profile and check their schedule.
6. **Export and Deployment:** Final schedules can be exported as CSV, PDF, or Excel files, or published to institutional systems.
7. **Advising:** The chatbot and advising tools automatically update recommendations based on the finalized timetable and student's academic plan.

This workflow ensures that MounouZinc integrates seamlessly into academic planning procedures and supports continuous refinement throughout the scheduling period.

4 Architecture and Implementation

While Section 3 introduces the conceptual components of MounouZinc, this section describes their technical realization and the interactions between layers. MounouZinc is implemented using four core technologies—PostgreSQL, MiniZinc, Python, and Node.js—each mapped directly to the architectural layers of the system (see Figure 3). The architecture is fully modular, allowing institutions to configure, extend, or replace components as scheduling policies evolve.

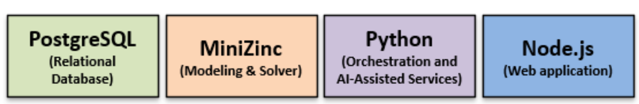

Fig. 3. Core Technologies

Figure 2 presents the high-level system architecture, organized into five layers: (4.1) data and persistence, (4.2) modeling and solver, (4.3) automation and orchestration, (4.4) web interface and interaction, and (4.5) AI-assisted services. Layers communicate through structured data schemas and APIs, ensuring separation of concerns and maintainability.

4.1 Data and Persistence Layer

The PostgreSQL database forms the foundation of the platform. Its relational schema includes tables for courses, professors, rooms, timeslots, departments, availability, workload, and academic plans. Foreign-key constraints enforce institutional consistency across departments.

4.2 Modeling and Solver Layer

Institutional data is mapped into MiniZinc decision variables, constraint modules, and optimization objectives. This modular design enables institutions and departments to activate only the constraint sets relevant to their specific rules, supporting flexibility and customization without altering the core engine. The modeling layer consists of:

1. *base domains* defining timeslots, professors, rooms, and timeslots;
2. *hard constraint modules* ensuring feasibility, including conflict prevention, room and professor availability;
3. *soft constraint modules* encoding preferences with adjustable weights for policies such as professor preferred teaching times.

4.3 Solver and Optimization Engine

MounouZinc uses MiniZinc as a solver-independent backend, supporting a variety of CP and hybrid solvers. Administrators can select the solver, adjust its time limits, and multi-objective weights reflecting institutional policy. Final solution is written back to the database and made available for review, visualization, and post-processing through the web interface. At this stage, the administrator can still perform changes on the generated timetable.

4.4 Automation and Orchestration Layer

The Python middleware automates the full data-to-solver pipeline. This layer ensures robust operation, reproducibility, and traceability across scheduling cycles, and enables integration with institutional workflows and external services. Responsibilities include:

1. generating MiniZinc data files from database content,
2. selecting and assembling appropriate constraint modules,
3. executing solvers and capturing logs,
4. parsing solution outputs, and
5. providing API endpoints for the web interface and external systems.

4.5 User Interface and AI Components

The professor–course matching module relies on transformer-based sentence embeddings to encode both course descriptions and faculty profiles into a shared semantic space. Similarity scores are computed using cosine similarity and used to rank candidate assignments. These scores are incorporated into the optimization model as weighted soft constraints, thereby influencing assignment decisions while preserving feasibility.

In parallel, the iAdvisor conversational agent leverages a Retrieval-Augmented Generation (RAG) architecture combining institutional regulations, academic catalogs, and student records. The system integrates rule-based validation with LLM-generated responses to ensure consistency with academic policies while providing flexible natural-language interaction.

5 Features and Usage

MounouZinc provides a comprehensive suite of features that support administrators, professors, and students throughout the scheduling lifecycle. Its modular design enables institutions to tailor the system according to their operational requirements while preserving a unified workflow.

5.1 Constraint Definition and Customization

Administrators can define and modify both hard and soft constraints through structured interfaces. Hard constraints (HC) must be strictly satisfied to ensure feasibility, while soft constraints (SC) are associated with weighted penalties, allowing fine-grained control over the overall quality of the generated schedule. All constraints are automatically translated into MiniZinc modules during model generation. Constraint activation is controlled through a dedicated database table, which specifies which modules should be included in a given MiniZinc model. This table is populated dynamically based on user input provided through the web application. The following list presents the different constraints considered:

1. HC: No two sessions can be scheduled in the same room at the same timeslot.
2. HC: Assigned professor must be qualified to teach the course.
3. HC: Assigned professor must be available during the session's timeslot.
4. HC: Professor must be assigned only one session in the same timeslot.
5. HC: Professors' total credit load must not exceed their maximum load.
6. SC: Prefer assigning courses to higher-priority timeslots.
7. SC: Prefer assigning courses to higher-priority rooms.
8. SC: Prefer assigning professors with higher suitability (course preference).
9. SC: Prefer assigning professors to their preferred timeslots.
10. SC: (Optional) Minimize total number of rooms used.

5.2 User Workflows

1. **Administrators** manage the timetabling system by importing data, maintaining core entities (professors, rooms, courses, timeslots), defining preferences and constraints, configuring the MiniZinc solver, running optimization, reviewing and updating the generated timetable if needed.
2. **Coordinators** manage departmental data by maintaining professors and courses, defining professor-course assignments, and verifying professors' time preferences.
3. **Professors** use a dedicated portal to specify their availability and preferred teaching slots. They can also view their assigned timetable, including courses, rooms, and teaching load, along with relevant statistics.
4. **Students** interact with the system through the iAdvisor chatbot to obtain information about degrees and majors and receive course recommendations based on prerequisites and their academic plan.

5.3 Visualization, Diagnostics, and Integration

MounouZinc provides an interactive timetable viewer that supports multiple perspectives, including overall schedule views, professor- and department-based views, and section-level views, complemented by charts presenting various statistical insights. All generated views can be exported in CSV format for further analysis and reporting. In addition, the system supports deployment either on-premise or within private cloud environments and integrates seamlessly with institutional SIS APIs, HR systems for managing professor data, and LMS platforms for schedule publication.This combined capability enables institutions to monitor, analyze, and distribute schedules efficiently while embedding MounouZinc into existing academic workflows with minimal operational overhead. These embedded features translate into stakeholder gains: administrators benefit from clearer constraint management, faculty members from a preference-entry portal and AI-assisted for availability, teaching wishes, expertise and students from an intelligent advising interface that guides their enrollment decisions and course selection.

6 Conclusion

This paper introduced MounouZinc, a comprehensive MiniZinc-based framework for university course timetabling that integrates optimization, data collection, recommendation, and conversational support into a single cohesive system. Unlike existing UCTP solutions—which typically focus solely on optimization or scheduling—MounouZinc extends functionality through its web interface for professor availability, AI-based professor–course matching, and a university chatbot capable of supporting prospective and current students. Experimental evaluation was conducted on synthetically generated datasets reflecting realistic institutional scenarios, ranging from small instances (3 professors, 5 sessions per week) to large-scale ones (44 professors, 243 sessions per week), MounouZinc consistently produced feasible, high-quality timetables, demonstrating both the scalability of the framework and the inherent hardness of large UCTP instances. Beyond raw performance, the evaluation highlighted concrete benefits for stakeholder groups. Institutions benefit from reproducible workflows, automated data pipelines, and integrated student-support services. Future work will focus on hybrid constraint programming and machine learning approaches and expanded constraint modules for cross-campus coordination, advanced LLM-based advising, and multi-objective metrics for workload distribution. MounouZinc aims to serve as both a practical institutional tool and a research platform that advances the state of the art in academic scheduling and educational decision-support systems.

References

1. A. Schaerf, "A Survey of Automated Timetabling," The Artificial Intelligence Review, vol. 13, no. 2, pp. 87–127, Apr. 1999, doi: 10.1023/A:1006576209967.
2. M. C. Chen, S. N. Sze, S. L. Goh, N. R. Sabar, and G. Kendall, "A Survey of University Course Timetabling Problem: Perspectives, Trends and Opportunities," IEEE Access, vol. 9, pp. 106515–106529, 2021, doi: 10.1109/ACCESS.2021.3100613.
3. H. Babaei, J. Karimpour, and A. Hadidi, "A survey of approaches for university course timetabling problem," Computers & Industrial Engineering, vol. 86, pp. 43–59, Aug. 2015, doi: 10.1016/j.cie.2014.11.010.
4. M. Mokhtari, M. Vaziri Sarashk, M. Asadpour, N. Saeidi, and O. Boyer, "Developing a Model for the University Course Timetabling Problem: A Case Study," Complexity, vol. 2021, pp. 1–12, Dec. 2021, doi: 10.1155/2021/9940866.
5. M. Davison, A. Kheiri, and K. G. Zografos, "Modelling and solving the university course timetabling problem with hybrid teaching considerations," J Sched, vol. 28, no. 2, pp. 195–215, Apr. 2025, doi: 10.1007/s10951-024-00817-w.
6. A. Muklason et al., "Flexible Automated Course Timetabling System with Lecturer Preferences Using Hyper-heuristic Algorithm: 7th International Conference on Sustainable Information Engineering and Technology, SIET 2022," SIET 2022 - Proceedings of 7th International Conference on Sustainable Information Engineering and Technology 2022, pp. 258–262, Nov. 2022, doi: 10.1145/3568231.3568273.
7. S. Cha, M. Loeser, and K. Seo, "The Impact of AI-Based Course-Recommender System on Students' Course-Selection Decision-Making Process," Applied Sciences, vol. 14, no. 9, p. 3672, Jan. 2024, doi: 10.3390/app14093672.
8. Jiawei Li, Qianru Lyu, Wei Qiu, and Andy W. H. Khong, "Improving Course Recommendation Systems with Explainable AI: LLM-Based Frameworks and Evaluations," July 2025, doi: 10.5281/ZENODO.15870185.
9. K. Peyton, S. Unnikrishnan, and B. Mulligan, "A review of university chatbots for student support: FAQs and beyond," Discov Educ, vol. 4, no. 1, p. 21, Jan. 2025, doi: 10.1007/s44217-025-00397-7.
10. S. Ceschia, L. D. Gaspero, and A. Schaerf, "Educational Timetabling: Problems, Benchmarks, and State-of-the-Art Results," European Journal of Operational Research, vol. 308, no. 1, pp. 1–18, July 2023, doi: 10.1016/j.ejor.2022.07.011.
11. N. M. Arratia-Martinez, P. A. Avila-Torres, and J. C. Trujillo-Reyes, "Solving a University Course Timetabling Problem Based on AACSB Policies," Mathematics, vol. 9, no. 19, Art. no. 19, Jan. 2021, doi: 10.3390/math9192500.

12. A. Gülcü and C. Akkan, "Robust university course timetabling problem subject to single and multiple disruptions," European Journal of Operational Research, vol. 283, no. 2, pp. 630–646, June 2020, doi: 10.1016/j.ejor.2019.11.024.
13. A. Ozkan, A. Ulucan, C. Dirik, and K. B. Atici, "University course timetabling with multi-section courses, room stability and professor preferences: an application in a business school," Comput Manag Sci, vol. 22, no. 1, p. 3, Feb. 2025, doi: 10.1007/s10287-025-00529-2.
14. Z. Akbar, E. Sopandi, B. Badruzzaman, and M. Khalik, "The Role of Artificial Intelligence-Based Recommendation Systems in Selection of Courses for Students," Journal of Social Science Utilizing Technology, vol. 1, pp. 249–260, Dec. 2023, doi: 10.70177/jssut.v1i4.671.
15. A. Kord, A. Aboelfetouh, and S. M. Shohieb, "Academic course planning recommendation and students' performance prediction multi-modal based on educational data mining techniques," J Comput High Educ, Jan. 2025, doi: 10.1007/s12528-024-09426-0.
16. M. Dawood, "Assessing the effectiveness of Chatbots in providing personalized academic advising and support to higher education students: A narrative literature review," Studies in Technology Enhanced Learning, vol. 4, no. 1, Oct. 2024, doi: 10.21428/8c225f6e.7140f8f4.
17. M. M. Thottoli, B. H. Alruqaishi, and A. Soosaimanickam, "Robo academic advisor: Can chatbots and artificial intelligence replace human interaction?," CONT ED TECHNOLOGY, vol. 16, no. 1, p. ep485, Jan. 2024, doi: 10.30935/cedtech/13948.
18. B. R. Ranoliya, N. Raghuwanshi, and S. Singh, "Chatbot for university related FAQs," in 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi: IEEE, Sept. 2017, pp. 1525–1530. doi: 10.1109/ICACCI.2017.8126057.