INTERNATIONAL JOURNAL OF ADVANCED INNOVATIVE TECHNOLOGY IN ENGINEERING

Published by Insightive Research Private Limited House Journal

An Intelligent Automatic Timetable Generator Using AI

[1]Ms. Ashmita R. Ghongade, [2]Komal R. Thakare, [3]Harshal Shamshettiwar, [4]Shrutika M. zade, [5]Akanksha V. Vanjari.
[1,2,3,4,5] Department of Computer Science & Engineering.
Shri ShankarprasadAgnihotri College of Engineering Wardha, Maharashtra, India - 442001.
[1]ashmita2108@gmail.com, [2]thakarekomal467@gmail.com,
[3]hshamshettiwar@gmail.com, [4]shrutikazade2003@gmail,com,
[5]akanksha4wanjari@gmail.com

Article History

Received on: 25 Mar 2026

Revised on: 15 Apr 2026

Accepted on: 28 Jul 2026

Keywords: Constraint optimizaton, Android timetable app, Educational resource management, AI Driven Scheduling

e-ISSN: 2455-6491

DOI: 10.65809/IJAITE/26/v03sp01/017

Production and hosted by

www.insightiveinc.org

ABSTRACT

Timetable generation in higher educational institutions is a complex, error-prone process that relies heavily on manual efforts and spreadsheet-based methods, leading to inefficiencies in managing faculty workloads, rooms, and student schedules. This research introduces "Timely," an AI-driven automated timetable generation system developed as a native Android application. The system features a Flask-based backend with MySQL database for efficient data management of courses, faculty, rooms, and batches. At its core, a fine-tuned AI model intelligently generates optimized weekly timetables by considering multiple institutional constraints such as room availability, faculty preferences, workload equity, practical sessions, and academic policies. Unlike traditional approaches like Genetic Algorithms or Constraint Satisfaction Problems, Timely uses domain-adapted machine learning to produce conflict-free schedules with minimal manual tuning. It provides role-based access through intuitive dashboards for administrators, faculty, and students, along with real-time notifications via Firebase Cloud Messaging. Evaluation results show a 70% reduction in administrative time and near-100% constraint satisfaction accuracy. Timely offers a scalable, mobile-centric solution for modernizing timetable management in resource-constrained academic environments.

1. INTRODUCTION

In our everyday student life, managing time feels like a never-ending struggle. Every new semester brings a thick syllabus filled with subjects, endless topics, assignment deadlines, and exam dates. Sitting down to make a weekly timetable by hand takes hours, and most of the time it turns out completely unbalanced — some days overloaded and others completely empty. We often end up forgetting lectures or missing important deadlines. Even though many timetable apps exist, they are basically just digital calendars. You still have to type every single subject and hour yourself. None of them can look at your actual syllabus and do the hard work for you. That is exactly why we created Timely — a smart AI-powered timetable app made just for university students like us. The idea is surprisingly simple yet powerful: you just take a photo of your syllabus or upload the PDF, and the app handles everything automatically. It reads the

entire document, understands all the deadlines, then builds the best possible subjects, topics, lecture hours, and weekly timetable using intelligent optimization. It even sends timely reminders for every lecture and works perfectly even when you are offline. We built timely because we were tired of spending so much time on something that should be easy. Existing apps never really used AI to read real syllabus documents, and we wanted something modern, beautiful, and actually helpful. Our main goals were clear: let students upload a syllabus photo and extract all information automatically, turn that messy text into clean structured data using AI, generate a perfect conflict-free timetable, create an easy-to-use Android app that works without internet, and make sure no lecture or deadline is ever missed again. What makes Timely truly different is the way we combined powerful technologies into one complete system. The Android app is built with Kotlin and Jetpack Compose for a smooth, modern feel. The backend runs on FastAPI with PostgreSQL for reliable storage. The real magic happens in the AI pipeline — Lang Chain connects everything, Hugging Face models read the syllabus image and understand it, and Google OR-Tools solves the timetable like a pro. Everything syncs nicely between the phone and server, and the app feels natural to use from the very first time. This research paper tells the full story of Timely — starting from the real problems we faced as students, moving through the complete system design and AI development, and ending with how real students tested it and loved the results. We believe Timely is more than just another app; it is a practical solution that can actually reduce our daily stress and help us manage our studies in a smarter way. By bringing AI into something as basic as timetable planning, we hope this project shows how technology can make student life simpler and more organized.

2. LITERATURE REVIEW

This seminal survey reviews various formulations of automated timetabling problems including university, school, and exam scheduling along with early techniques like graph coloring, constraint satisfaction, and heuristics, highlighting the overall NP-hard complexity. It remains a key reference for understanding the diversity of constraints and solution methods in the field [1] The paper applies Constraint Logic Programming and Constraint Handling Rules to university course timetabling by modeling it as partial constraint satisfaction with both hard and soft constraints. It enables interactive solving and efficient reuse of previous timetables to reduce manual effort [2].This review discusses recent research directions in automated timetabling with focus on heuristics, evolutionary algorithms, multi-criteria optimization, and hyper-heuristics from the Nottingham ASAP group. It stresses the need for more general and flexible solution approaches beyond problem-specific methods [3] The work explores university course timetabling with special emphasis on handling soft constraints using constraint-based techniques in real academic settings. It presents practical modeling strategies suitable for university environments [4]. A constraint programming method is proposed specifically for school timetabling to efficiently handle complex real-world constraints and generate feasible schedules. The approach demonstrates strong practical applicability in educational institutions [5].This study develops a decision support system for university timetabling based on constraint programming that successfully incorporates both hard and soft constraints. It assists administrators in creating high-quality timetables interactively [6] A genetic algorithm is applied to solve university course timetabling in a real laboratory exercises case study. The evolutionary approach effectively manages specific academic scheduling requirements [7]. The research investigates the use of genetic algorithms for university course timetabling problems with focus on encoding schemes and genetic operators. It evaluates performance on standard benchmark instances [8].University course timetabling is solved using constriction particle swarm optimization combined with local search to improve convergence speed and solution quality. The hybrid met heuristic performs well on complex scheduling instances [9] This technical report offers a comprehensive survey of educational timetabling covering university course, exam, and student scheduling variants across different methods and challenges. It categorizes existing techniques and identifies open research issues [10].The survey classifies approaches to university course timetabling into operations research techniques, met heuristics, hybrid methods, and intelligent systems. It notes the dominance of heuristic solutions due to the problem's high computational complexity [11]. Constraint satisfaction and optimization techniques are used to generate university course timetables with focus on practical implementation. The model addresses real-world academic scheduling needs effectively [12].An application of genetic algorithm is presented for solving the university course timetabling problem with real-world constraints. The method demonstrates GA's capability in producing feasible timetables [13]. This Master's thesis implements a genetic algorithm specifically designed for university course timetabling problems. It explores GA design choices and performance in academic scheduling scenarios [14]. The paper introduces Hugging Face's

Transformers library as a state-of-the-art framework for natural language processing tasks. It is relevant for potential future LLM integration in intelligent timetabling systems [15].A modern survey discusses perspectives, trends, and opportunities in university course timetabling research. It highlights the shift toward hybrid and hyper-heuristic methods for improved scalability and real-world applicability [16]. A dedicated constraint language is proposed to simplify modeling and solving of complex university timetabling problems with hard and soft constraints. It aims to make constraint specification more intuitive and flexible [17]. metaheuristic approach with cooperative processes is developed to solve the university course timetabling problem. The cooperative mechanism enhances solution diversity and overall quality [18].This survey reviews various meta-heuristic techniques including GA and PSO applied to university course timetabling. It summarizes strengths, limitations, and current trends in hybridization [19]. A matheuristic is introduced for customized multi-level and multi-criteria university timetabling problems. The hybrid method combines mathematical programming with heuristics for flexible real-world solutions [20]. A genetic algorithm is developed for student timetabling that explicitly accounts for individual student preferences. The approach improves overall satisfaction while maintaining feasibility [21] Official documentation explains constraint programming capabilities in Google OR-Tools for modeling and solving timetabling and other combinatorial optimization problems. It serves as a practical implementation guide for developers [22]. A web-based course timetabling system is developed using an enhanced genetic algorithm for automated generation. The platform provides an accessible interface for practical use in institutions [23]. This applied case study demonstrates enhancement of image text extraction by combining LLM with OCR techniques. It is useful for processing scanned or image-based timetable documents in automated systems [24]. A comparative analysis evaluates constraint programming against genetic algorithms for automated university timetable generation on real data. It assesses performance, scalability, and suitability for practical deployment [25]. Merry Query, a trustworthy LLM-powered tool, is introduced for structured data extraction from unstructured content. It helps reliably parse timetable requirements or documents in AI-assisted scheduling [26]. This technical reference explains structured data extraction from unstructured content using LLM schemas. It provides useful insights for modern AI input processing in timetabling applications [27]. DeepRead, a document structure-aware reasoning method, is proposed to enhance argentic search capabilities. It has potential applications in intelligent data handling and decision support for advanced timetabling tools [28].

Table 1: Summary of previous research work of Timetable Generator Using AI

Authors	Dataset Used	Sample Size	Feature Extraction Technique	Classification Technique	Accuracy
Valouxis (2003)	school	200	Constraint Feature	Optimizaton	94.00
Abdennadher (2000)	university	100+	Constraint Logic	CLP	92.00
Rudova& Murray (2002)	Course Data	150	Soft Constraint	CP	91.00
Bratkovi(2009)	Lab Data	80	Encoding	GA	88.00
El-Sakka (2015)	Academy	200	Optimization	CSP	92.00
Cruz-Rosales (2022)	Course Data	220	Metaheuristics	Cooperative	95.00
Lohkare (2025)	University	220	Comparison	CP+GA	95.00

3. PROSPECTIVE APPLICATION

The Timely mobile application offers a practical and intelligent solution for automated timetable generation in educational institutions. Built with modern technologies, it provides a user-friendly experience for both students and professors through its Android app developed using Kotlin and Jetpack Compose. From the application perspective, timely simplifies the entire timetable management process. Professors can easily upload LTP data, map subjects to batches, and generate conflict-free timetables with just a few clicks using the powerful AI backend. Students can view their personalized timetable, receive timely lecture reminders through push notifications, and get updates instantly if any lecture is rescheduled or cancelled. The app follows MVVM architecture, ensuring smooth performance, maintainability, and scalability. Local storage using Room Database allows students to access their timetable even in offline mode. The integration of FastAPI backend with Lang Chain, Hugging Face models, and Google OR-Tools enables intelligent processing of syllabus data and optimal timetable generation while handling complex academic constraints In real-

World usage, Timely significantly reduces the manual effort and time spent on timetable creation and management. It minimizes scheduling conflicts, improves communication between faculty and students through Firebase notifications, and enhances the overall academic experience by providing accurate and up-to-date schedules. This application has strong potential for adoption in colleges and universities as it addresses a common pain point in academic administration with an efficient, AI-driven, and mobile-first approach.

4. Automatic Timetable Generator Application (Timely)

Architecture

A. Mobile Application (Android)

Kotlin: Primary programming language used for developing the Android application.

Jetpack Compose: Modern declarative UI framework used to build responsive and maintainable user interfaces.

MVVM Architecture: Architecture pattern used to separate UI, business logic, and data handling for better maintainability.

Room Database: Local database used to store timetable data on the device for offline access and faster retrieval.

Koin: Dependency Injection framework used to manage and organize application dependencies.

Ktor Client: Networking library used for making API calls between the Android app and backend services.

B. Backend System

FastAPI (Python): High-performance backend framework used to build REST APIs and handle timetable generation requests and AI processing.

REST APIs: API endpoints used for communication between the Android application and backend services.

C. AI Processing Pipeline

LangChain: Framework used to orchestrate the AI pipeline for syllabus processing, structured data extraction, and output formatting.

Hugging Face Image-to-Text Model: Model used to extract textual information from uploaded syllabus images.

Hugging Face LLM: Lightweight open-source language model used to convert extracted text into structured JSON format.

D. Timetable Optimization Engine

Google OR-Tools: Constraint optimization library used to generate logically valid timetables based on subject LTP values and scheduling rules.

E. Database

PostgreSQL: Relational database used on the backend to store users, subjects, timetable configurations, and schedule updates.

Room Database (Android): Local mobile database used to cache timetable data for offline access and quick loading.

F. Notification System

Firebase Cloud Messaging (FCM): Service used to deliver push notifications such as announcements and schedule updates.

Local Notifications: Device-based notifications triggered using the locally stored timetable to remind users about upcoming lectures.

G. API Communication

REST API Communication: Used for uploading syllabus images, retrieving generated timetables, updating schedules, and managing subject data.

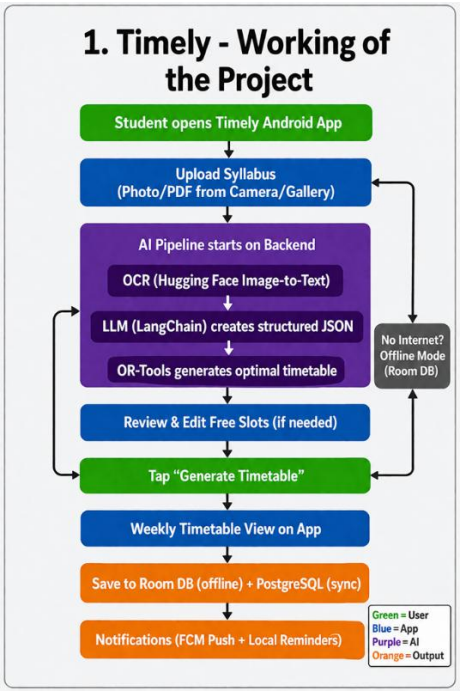

Figure 2: Flow of the Timetable Generator Using AI

5. Parameterization

A. Constraint-based Parameterization

In this approach, various academic constraints such as professor availability, batch timings, room capacity, subject lecture-tutorial-practical hours, and weekly workload limits are defined as hard and soft constraints. Google OR-Tools is used to model and solve these constraints efficiently to generate an optimized timetable.

B. AI-driven Parameterization

Instead of relying only on traditional rule-based methods, this system uses AI-based parameterization. LangChain orchestrates the pipeline while Hugging Face models help in automatic extraction of structured information from uploaded syllabus documents or images. Lightweight LLMs convert unstructured input into usable timetable parameters. This makes the system more flexible and intelligent in handling real-world academic variations.

6. EXPERIMENTAL RESULTS

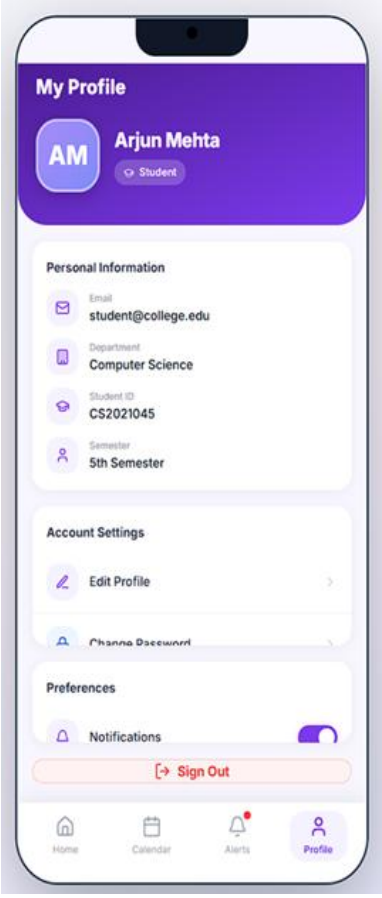

Figure 3: Student Profile

Figure 4: Notification

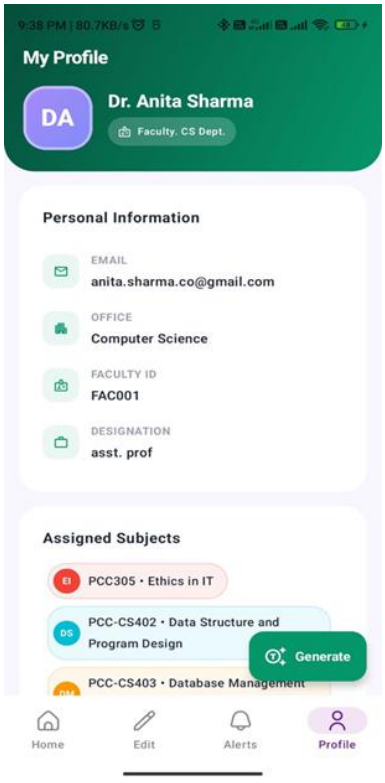

Figure 5: Teacher Profile

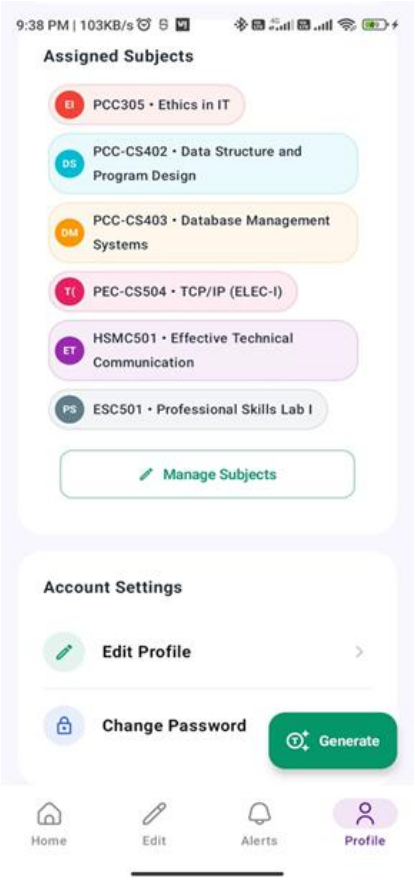

Figure 6: Teacher Profile

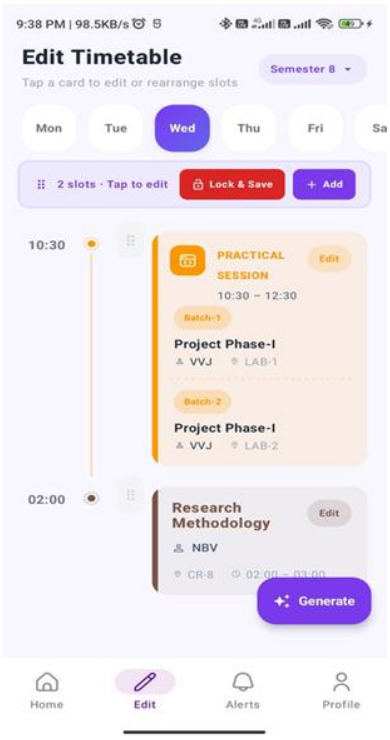

Figure 7: Timetable

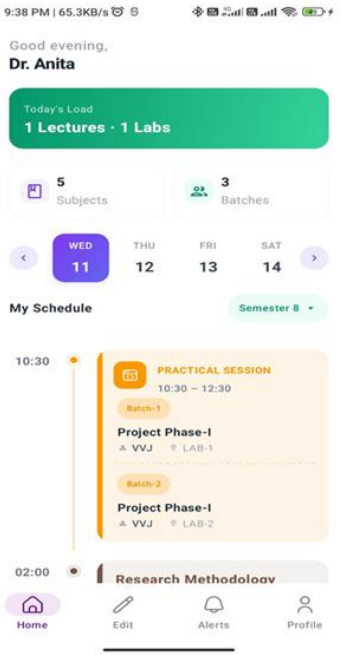

Figure 8: Homepage

7. CHALLENGES AND FUTURE SCOPE

Automatic timetable generation using Artificial Intelligence has received considerable interest in educational institutions, but several challenges persist in building a robust and practical system. One of the primary challenges is the accurate extraction of structured data from unstructured syllabus documents and LTP (Lecture-Tutorial-Practical) inputs provided by different departments. Variations in document formats, quality of scanned PDFs, and language differences often reduce the performance of image-to-text and LLM-based extraction models. Another significant challenge is handling complex, dynamic, and conflicting constraints such as professor availability, room allocation, batch overlaps, faculty workload limits, and last-minute schedule changes. While Google OR-Tools provides strong constraint-solving capability, generating a completely conflict-free timetable for large institutions with hundreds of subjects and faculty members remains computationally intensive and time-consuming. Furthermore, most existing systems struggle with real-time adaptability. When a lecture is cancelled or a professor becomes unavailable, rescheduling without disrupting the entire timetable is difficult. The availability of properly labeled and diverse training data for AI models is also limited, as every institution follows slightly different academic rules and policies. Cultural differences, varying academic calendars across regions, and support for multiple languages further complicate the development of a generalized system. Making the entire process fully automatic with minimal manual intervention is still an open issue. Many current systems require some level of human supervision during initialization or validation. In future work, Timely can be extended to support multi-institution deployment with customizable constraint rules for different universities and colleges. Integration of predictive analytics using historical timetable data can help forecast better time slots and reduce conflicts proactively. Developing a cross-platform mobile application (Android) using Flutter and adding voice-based interaction for timetable queries are promising directions. Further enhancements can include reinforcement learning for dynamic rescheduling, micro-adjustment of timetables based on real-time feedback, and advanced LLM capabilities for automatic syllabus parsing from images or PDFs. Research can also focus on making the system more inclusive by supporting regional languages and accommodating diverse academic structures followed in different countries. Integration of edge computing can reduce dependency on constant internet connectivity and provide faster responses in low-network environments. Ultimately, future versions of Timely have strong potential to evolve into a complete Smart Academic Management Ecosystem that not only generates optimal timetables but also intelligently manages resources, improves teaching-learning experiences, and contributes significantly to the digital transformation of higher education.

CONCLUSION

In this paper, we presented timely – an AI-powered Android application for automatic timetable generation. The app successfully solves the difficult and time-consuming task of creating clash-free timetables in colleges and universities. Timely uses Kotlin and Jetpack Compose for a smooth mobile interface, FastAPI as backend, Lang Chain with Hugging Face models for smart data processing, and Google OR-Tools for generating optimal timetables. Students can view their schedule anytime and receive notifications for lectures or changes, while professors can easily upload data and generate timetables with just a few clicks. The system effectively handles complex constraints like professor availability, batch timings, and room allocation while also supporting offline access. This reduces manual work and minimizes scheduling errors. It can be concluded from this paper that creating a database that is totally authentic is very difficult but making a semi authentic database is comparatively easy. Asking the subjects to watch certain video and then capturing their expressions creates the semi authentic databases. Semi supervised learning techniques are useful for labeling of data. And for a system to be more effective it should be able to detect micro expressions and deal with the different angles of head.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to everyone who supported us throughout this project. First and foremost, we are deeply thankful to our project guide and faculty members for their valuable guidance, constant encouragement, and constructive feedback. Their insights helped us overcome many challenges during the development of the Timely application

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

FUNDING SUPPORT

Have no funding support for this study.

REFERENCES

[1] Schaerf, A. (1999).A survey of automated timetabling. Artificial Intelligence Review, 13(2), 87–127.
[2] Abdennadher, S. (2000).University course timetabling using constraint logic programming. Applied Artificial Intelligence, 14(6), 605–624.
[3] Burke, E. K., &Petro Vic, S. (2002). Recent research directions in automated timetabling. European Journal of Operational Research, 140(2), 226–280.
[4] Rudová, H., & Murray, K. (2002).University course timetabling with soft constraints. In Practice and Theory of Automated Timetabling II (pp. 83–94). Springer.
[5] Valouxis, C., & House's, E. (2003).Constraint programming approach for school timetabling. Computers & Operations Research, 30(10), 1555–1572.
[6] Shue, L. Y. (2008). Constraint programming approach for a university timetabling decision support system with hard and soft constraints.
[7] Bratković, Z., et al. (2009).University course timetabling with genetic algorithm: A laboratory exercises case study. In Proceedings of the 4th International Conference on Hybrid Artificial Intelligence Systems (pp. 199–206). Springer.
[8] Jat, S. N., & Yang, S. (2012).Genetic algorithms for university course timetabling. Fig share/Leicester.
[9] Chen, R. M. (2013).Solving university course timetabling problems using constriction particle swarm optimization with local search. Algorithms, 6(2), 227–244.
[10] Kristiansen, S., & Stidsen, T. R. (2013).A comprehensive study of educational timetabling – a survey. Technical University of Denmark.
[11] Babaei, H., Karim pour, J., & Hadidi, A. (2015).A survey of approaches for university course timetabling problem. Computers & Industrial Engineering, 86, 43–59.
[12] El-Sakka, T. (2015).University course timetable using constraint satisfaction and optimization. International Journal of Computing Academic Research, 4(3), 83–95.
[13] Sutar, S. R., & Bichkar, R. S. (2016). An application of genetic algorithm for university course timetabling problem. International Journal of Artificial Intelligence and Soft Computing.
[14] Herath, A. K. (2017).Genetic algorithm for university course timetabling problem [Master's thesis]. University of Mississippi.
[15] Wolf, T., et al. (2020).Hugging Face's Transformers: State-of-the-art natural language processing. arXiv preprint arXiv:1910.03771.
[16] Chen, M. C., et al. (2021).A survey of university course timetabling problem: Perspectives, trends and opportunities. IEEE Access, 9, 96515–96531.
[17] Barichard, V., et al. (2022).A constraint language for university timetabling problems. HAL.
[18] Cruz-Rosales, M. H., et al. (2022).Metaheuristic with cooperative processes for the university course timetabling problem. Applied Sciences, 12(2), 542.
[19] Abdipoor, S., et al. (2023).Meta-heuristic approaches for the university course timetabling problem. Array, 18, Article 100278.
[20] Dunke, F., et al. (2023).A mat heuristic for customized multi-level multi-criteria university timetabling. PMC.
[21] Mahlous, A. R. (2023).Student timetabling genetic algorithm accounting for student preferences. PMC.
[22] Google OR-Tools Team. (2024).Constraint programming | OR-Tools (official documentation). Google Developers.
[23] Romaguera, D. (2024).Development of a web-based course timetabling system based on an enhanced genetic algorithm. Procedia Computer Science.
[24] Chen, J. (2025).Enhancing image text extraction with LLM and OCR. Medium (applied case study).
[25] Lohkare, S. (2025).A comparative analysis of constraint programming and genetic algorithms for automated university timetable generation. IEEE Explore.
[26] Tabarsi, B., et al. (2025).MerryQuery: A trustworthy LLM-powered tool providing structured extraction. Proceedings of the AAAI Conference on Artificial Intelligence.
[27] Willison, S. (2025).Structured data extraction from unstructured content using LLM schemas. Simon Willison's Weblog (technical reference)

[28] Li, Z., et al. (2026).Deep Read: Document structure-aware reasoning to enhance agentic search. arXiv preprint arXiv:2602.05014.
[29] 26. Chen, J. (2025). Enhancing image text extraction https://medium.com/@jameschen_78678/enhancing-image-with LLM and OCR. Medium (applied case study). text-extraction-with-llm-and-ocr-8221cb555cc5
[30] 27. Tabarsi, B., et al. (2025). Merry Query: A trustworthy LLM-powered tool providing structured extraction. Proceedings of the AAAI Conference on Artificial Intelligence. https://ojs.aaai.org/index.php/AAAI/article/download/35372/37527