



International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

ClassHarmonica: Adaptive AI-Optimized Timetable Orchestrator

¹*Sudarshan Kadam, ¹Sakshi Palekar, ¹Atharva Pujari, ¹Sanket Malegaonkar, ²Prof. S. R. Ganorkar*

¹ Student, Department of Information Technology, Sinhgad College of Engineering, Pune, India.

² Head of Department, Department of Information Technology, Sinhgad College of Engineering, Pune, India.

ABSTRACT :

Timetable scheduling is one of the most essential and challenging tasks in educational institutions. Traditional methods rely prominently on manual coordination, leading to scheduling conflicts, inefficient resource allocation and problems in adapting to sudden changes. This paper presents ClassHarmonica: Adaptive AI-Optimized Timetable Orchestrator, an intelligent system designed to automate academic timetable generation using Artificial Intelligence (AI) and constraint optimization techniques. The system integrates Google's OR-Tools CP-SAT solver for constraint satisfaction with OpenAI's natural language model for intelligent rescheduling through user interaction. It is implemented as a web-based platform using Python and Flask. ClassHarmonica generates conflict-free, adaptive and multi-view timetables for faculty, divisions and classrooms. The system thus articulates AI and optimization techniques to deliver a scalable and practical solution for academic institutions like engineering colleges.

Keywords: Timetable Scheduling, Artificial Intelligence, Constraint Programming, OR-Tools, Optimization, Educational Automation.

1. Introduction

Efficient academic scheduling is important for smooth institutional operations. In engineering colleges, where curricula involve theory lectures, laboratory sessions, and tutorials across multiple years and divisions, timetable preparation becomes challenging complex. Manual methods are time-consuming, sensitive to conflicts and difficult to adapt when changes arise.

The timetable generation problem is a well-known Constraint Satisfaction Problem (CSP), where various variables such as subjects, faculty, rooms and time slots must satisfy institutional and logical constraints. Solving this problem manually is not feasible for large institutions due to its NP-hard nature. Hence, an intelligent automated system is essential to ensure accuracy, efficiency and adaptability.

The proposed system, ClassHarmonica, automates the process of timetable generation using constraint optimization and artificial intelligence. It ensures conflict-free scheduling, dynamic adaptability and ease of use through natural language interaction by users. The project addresses the academic, administrative and technological challenges in timetable generation and provides a scalable, real-world solution.

2. Literature Review

The timetabling problem has been extensively studied using various optimization and heuristic methods. Early approaches such as Genetic Algorithms (GA) and Simulated Annealing (SA) offered approximate solutions but lacked scalability.

Table 1 – Literature survey

Sr. No.	Author(s)/Year	Method Used	Key Idea	Limitation/Gap
1	S. Yang & S. N. Jat (2011)	Genetic Algorithm	Used local search for course	Needs parameter tuning, not dynamic
2	M. Cupic et al. (2009)	GA for Exam Scheduling	Handled room and time conflicts	Focused only on exams
3	E. Aycaan & T. Ayav (2009)	Simulated Annealing	Improved convergence for scheduling	Sensitive to parameter changes

4	UniTime (Open-Source Tool)	Constraint Programming	Open-source timetable system	Complex setup, no NLP support
---	----------------------------	------------------------	------------------------------	-------------------------------

3. System Design

The system architecture of ClassHarmonica is modular and follows a layered approach consisting of the Presentation Layer, Application Layer, Data Layer, and External Services Layer.

3.1 System Overview

- Frontend: Built using HTML5, CSS3, JavaScript, and Bootstrap for responsiveness and accessibility.
- Backend: Developed using Flask (Python) with integration of OR-Tools and OpenAI GPT-4o API.
- Data Storage: Utilizes JSON-based configuration with planned migration to PostgreSQL for persistence.
- AI and Optimization: OR-Tools CP-SAT solver generates conflict-free schedules; GPT-4o provides natural language interaction for modifications.

3.2 Core Modules

- Data Management Module – Handles academic structures (years, divisions, batches) and maintains data consistency.
- Constraint Modelling Module – Defines and enforces scheduling constraints (faculty, rooms, student overlap).
- Timetable Generation Module – Uses the CP-SAT solver to produce optimized, conflict-free schedules.
- AI-Assisted Interaction Module – Allows users to modify timetables using natural language commands.
- Visualization and Reporting Module – Generates multiple timetable views and validation reports.
- Adaptability Module – Handles emergency schedule updates with minimal disruption.

3.3 Component Interaction Flow

The workflow of the system can be described as follows:

1. **Input Stage:**
 - Administrator enters academic structure, subjects, faculty, and rooms.
 - Coordinator triggers timetable generation or requests modification.
2. **Processing Stage:**
 - Constraint Modelling Module encodes rules.
 - OR-Tools Solver generates an optimized schedule.
 - AI Assistant processes natural language requests for rescheduling.
3. **Output Stage:**
 - Timetable produced in multiple perspectives (division, faculty, room).
 - Validation report generated (conflicts, workload distribution, room utilization).
 - Users access timetables through the web interface.

The architecture of **ClassHarmonica** is modular and layered, ensuring scalability, adaptability, and user-friendliness. Each component plays a crucial role:

- The **frontend** ensures accessibility.
- The **application layer** provides the core scheduling logic and AI assistance.
- The **data layer** ensures accurate storage and retrieval of academic information.
- The **external services** bring optimization and intelligence to the system.

This design makes ClassHarmonica capable of handling the complexities of academic timetabling while remaining flexible and responsive to institutional needs.

3.4 Architectural Diagram

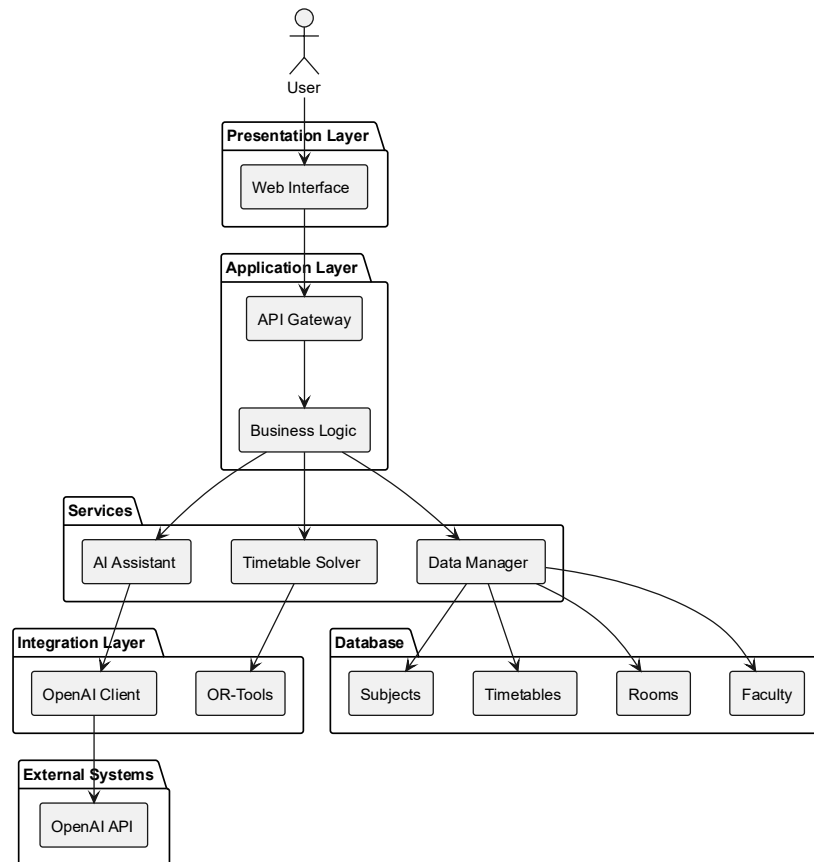


Fig. 1 – System Layered Architecture

4. Results and Discussion

At the current stage of development, the ClassHarmonica project has achieved completion of the problem identification, literature survey, system design, and around 30% of the coding and module integration. The initial prototype has been developed to validate the concept and architecture proposed in the design phase.

4.1 Preliminary Results

The partial implementation focuses on:

- **User Interface and Data Input Module:** Basic web interface created using Flask and HTML for entering faculty, subjects, and room data.
- **Constraint Definition Module:** Preliminary structure for hard constraints (faculty availability, room occupancy) has been coded and tested using sample data.
- **Prototype Schedule Generation:** A simplified version of the OR-Tools CP-SAT model was used to generate basic non-conflicting timetables for limited data (2 divisions, 5 faculty members).

4.2 Observed Outcomes

- The initial prototype successfully generates basic timetables without overlap in time slots.
- Constraint definitions and database structures were validated for scalability.
- The front-end data entry and JSON-based data storage performed reliably for smaller datasets.
- Early integration with OR-Tools confirmed the feasibility of constraint-based optimization for full-scale deployment.

4.3 Discussion

The progress so far validates the conceptual soundness and technical feasibility of the proposed system architecture. The implemented modules demonstrate that constraint-based optimization can effectively automate timetable generation when integrated with AI-driven adaptability in future iterations.

Upcoming work will focus on:

- Completing the AI-assisted interaction module for natural language-based rescheduling.
- Implementing dynamic adaptability for real-time schedule modifications.
- Conducting comprehensive testing with real institutional datasets.

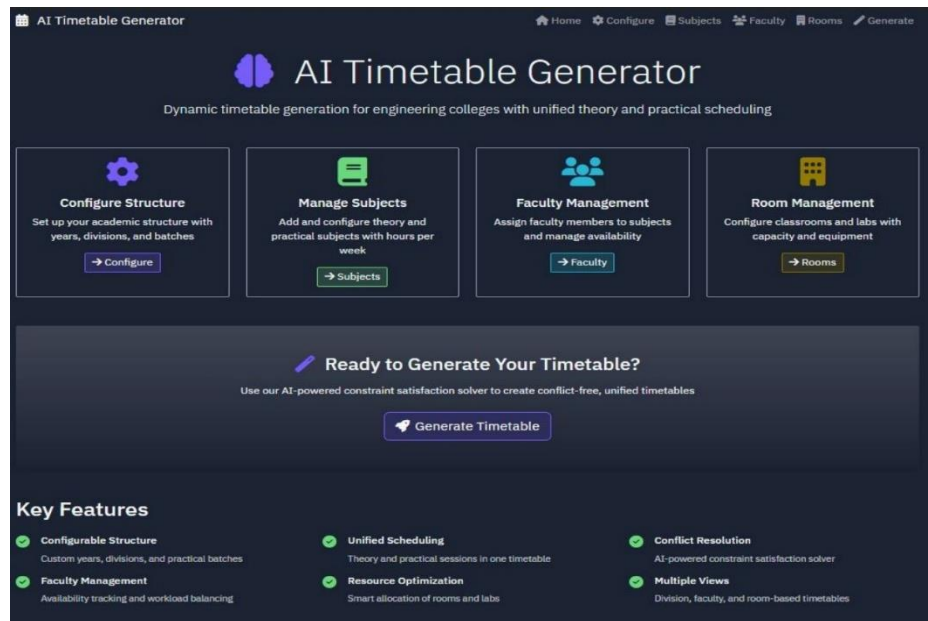


Fig. 2 – User Interface

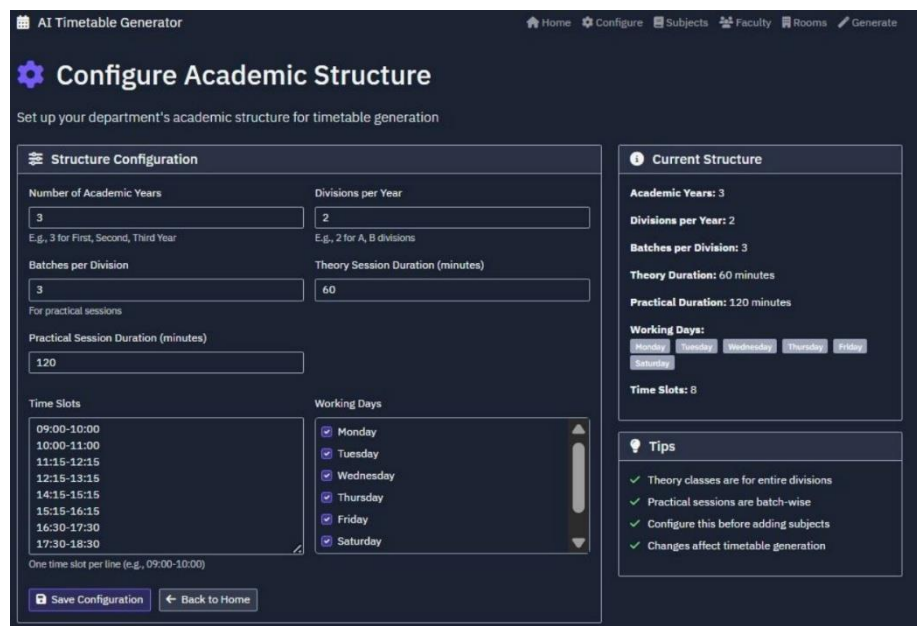


Fig. 3 – Configuration of Academic Structure

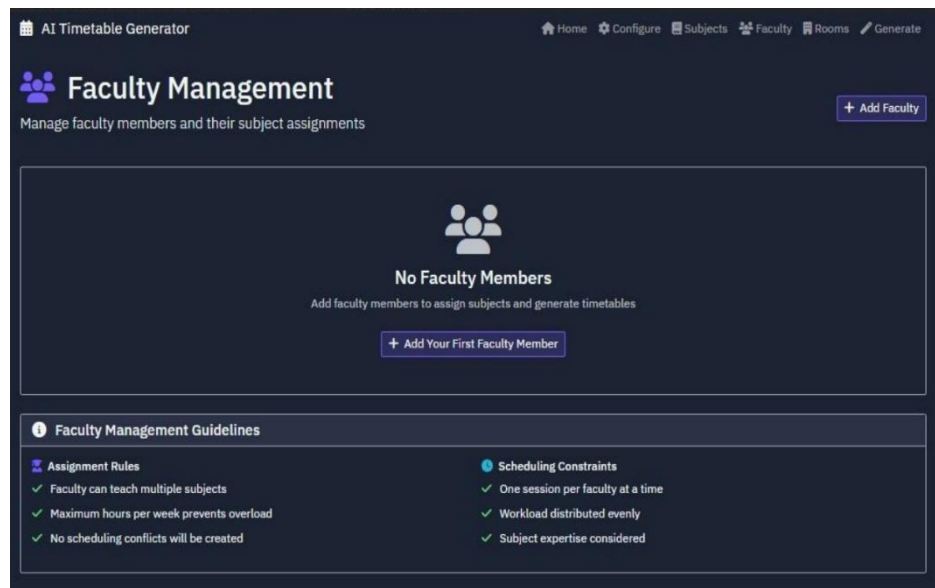


Fig. 4 – Faculty Management

5. Conclusion

ClassHarmonica: Adaptive AI-Optimized Timetable Orchestrator shows how artificial intelligence and constraint programming has changed the way we think about academic scheduling. The system automates complex timetabling tasks, reduces human error and adapts to changing conditions in real time. Its design enables efficient constraint satisfaction, intelligent user interaction and scalability. Future enhancements include migrating to a relational database for larger data handling, integrating machine learning for predictive scheduling and expanding AI features for more natural, context-aware interaction. Overall, ClassHarmonica offers a robust, intelligent and future-ready solution for academic institutions.

6. REFERENCES

- [1] S. Yang and S. N. Jat, "Genetic algorithms with guided and local search strategies for university course timetabling," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 1, pp. 93–106, Jan. 2011.
- [2] M. Cupic, M. Golub, and D. Jakobovic, "Exam timetabling using genetic algorithm," in *Proc. 31st Int. Conf. Information Technology Interfaces (ITI)*, Cavtat, Croatia, 2009, pp. 357–362.
- [3] E. Aycan and T. Ayav, "Solving the course scheduling problem using simulated annealing," in *IEEE Int. Advance Computing Conf. (IACC)*, Patiala, India, 2009, pp. 1–6.
- [4] Google, "OR-Tools Documentation." [Online]. Available: <https://developers.google.com/optimization>.
- [5] OpenAI, "OpenAI API Documentation." [Online]. Available: <https://platform.openai.com/docs>