AI-Based Course Timetable Generator

A project by:
Anna Mariya Martin - 24WU0101077
Hasitha Kocherla - 24WU0101078
Akkala Harshitha - 24WU0101088
P. Ramya Sai - 24WU0101090

Course ID: 24WU03MJM2

Submitted to:
Dr. Rajkumar Kalimuthu
Associate Professor
School of Technology
Woxsen University.

Submission Date: 14/10/2025

AI-Based Course Timetable Generator

Prepared by:

Anna Mariya Martin – 24WU0101077

Hasitha Kocherla – 24WU0101078

Akkala Harshitha – 24WU0101088

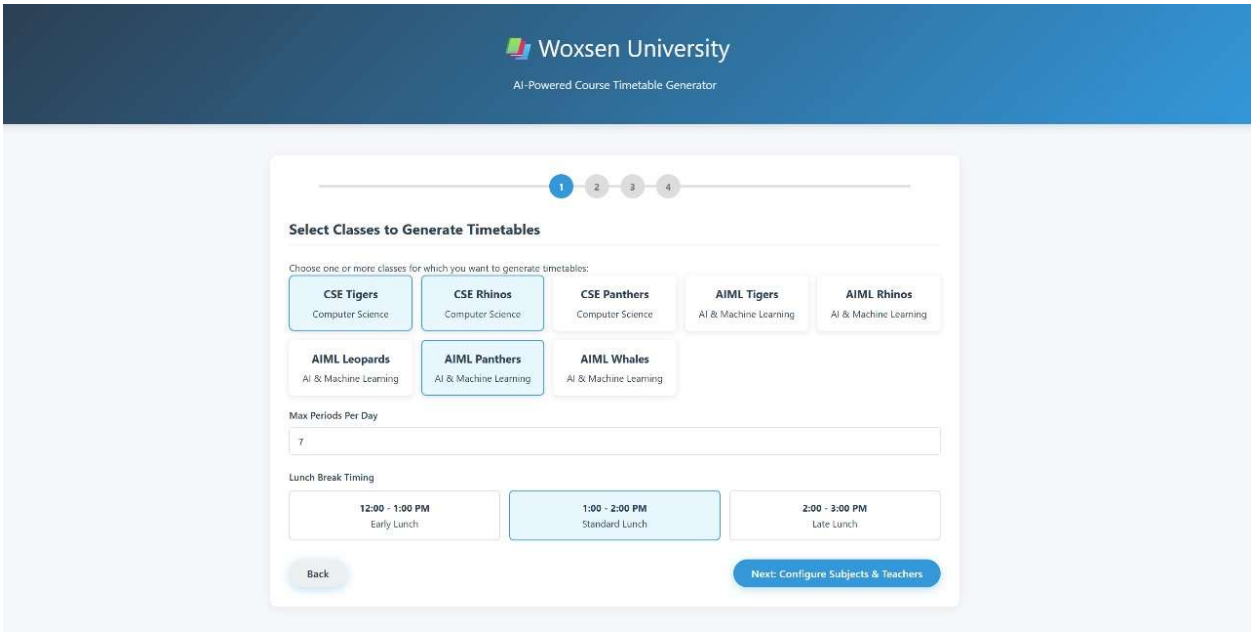
P. Ramya Sai – 24WU0101090

Objective and Problem Definition

The AI Timetable Generator project aims to automate the creation of academic timetables to optimize scheduling while preventing conflicts like teacher overlaps and room double bookings. This is a significant challenge in many educational institutions due to the complexity of balancing numerous constraints such as staff availability, classroom capacity, lunch breaks, and multi-class scheduling. The manual process of scheduling is labor-intensive and error-prone, often resulting in inefficient allocations and clashes. Our project delivers a user-friendly web-based solution that integrates advanced scheduling algorithms with real-time validation and export functionalities. The expected outcome is a robust platform that simplifies academic timetable management for administrators and enhances academic workflow efficiency.

System Architecture

- Modular web application with clearly defined components
- User input interface for classes, subjects, teachers, and rooms
- Scheduling engine running graph-coloring-based algorithm for conflict-free slot allocation
- Validation module to detect teacher and room scheduling conflicts
- Results module presenting timetable views by class, teacher, or room
- Export functionality for PDF, Excel, and Google Calendar integration
- Responsive UI updated in real-time based on user interactions and scheduling status



Techniques and Algorithms

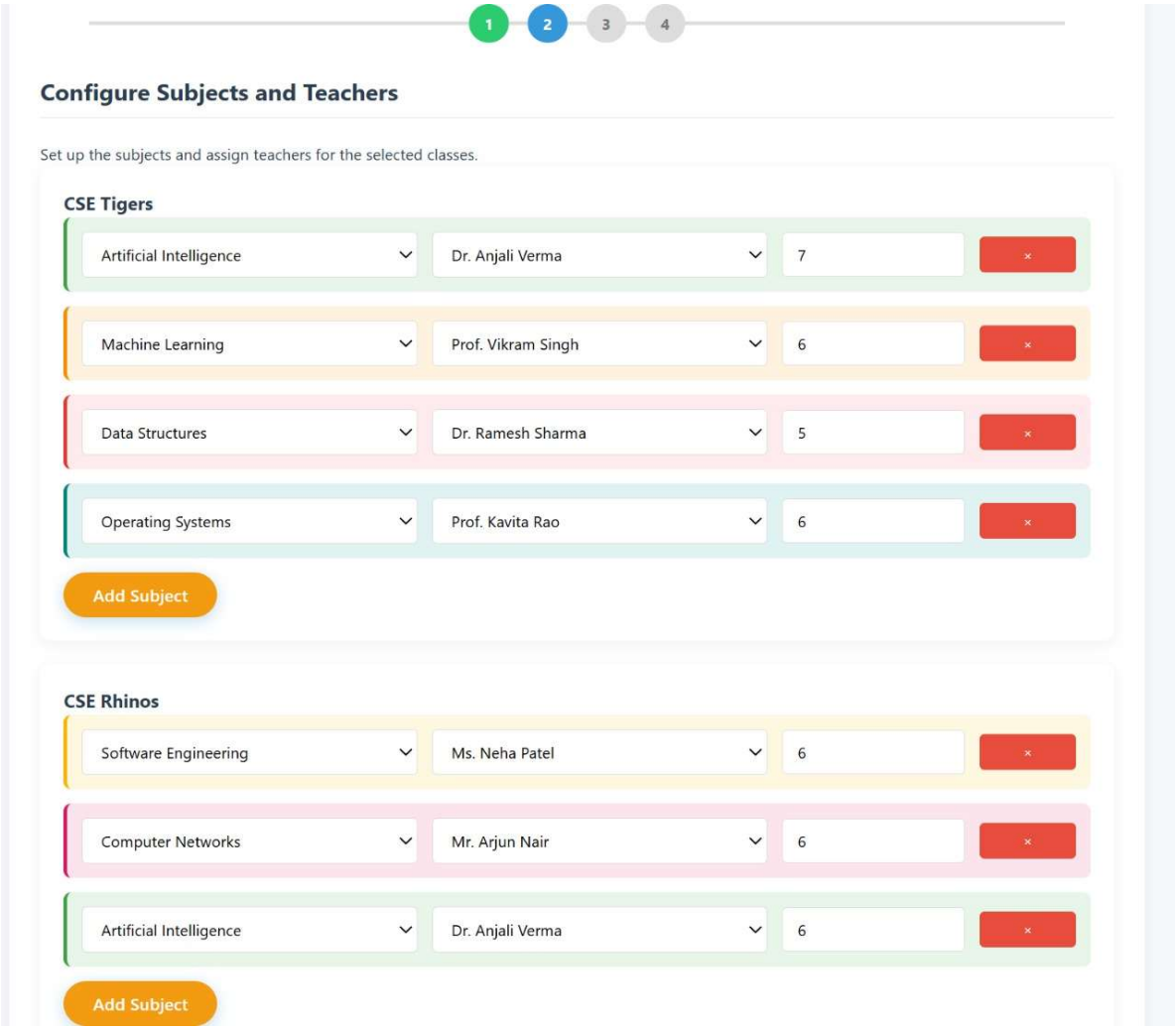
The project largely employs a graph coloring-based algorithm, a traditional computer science method appropriate for constraint satisfaction problems like scheduling. It corresponds classes, teachers, and classrooms as nodes and edges and assigns time slots systematically so as not to conflict. The rationale for choosing graph coloring is its tested efficiency and competency in various forms of scheduling problems. In addition to that, real-time validation logic provides users with instant feedback about input errors or scheduling conflicts. The project also includes third-party libraries such as jsPDF and xlsx for exporting timetable data, combining these libraries in order to offer complete platform functionality.

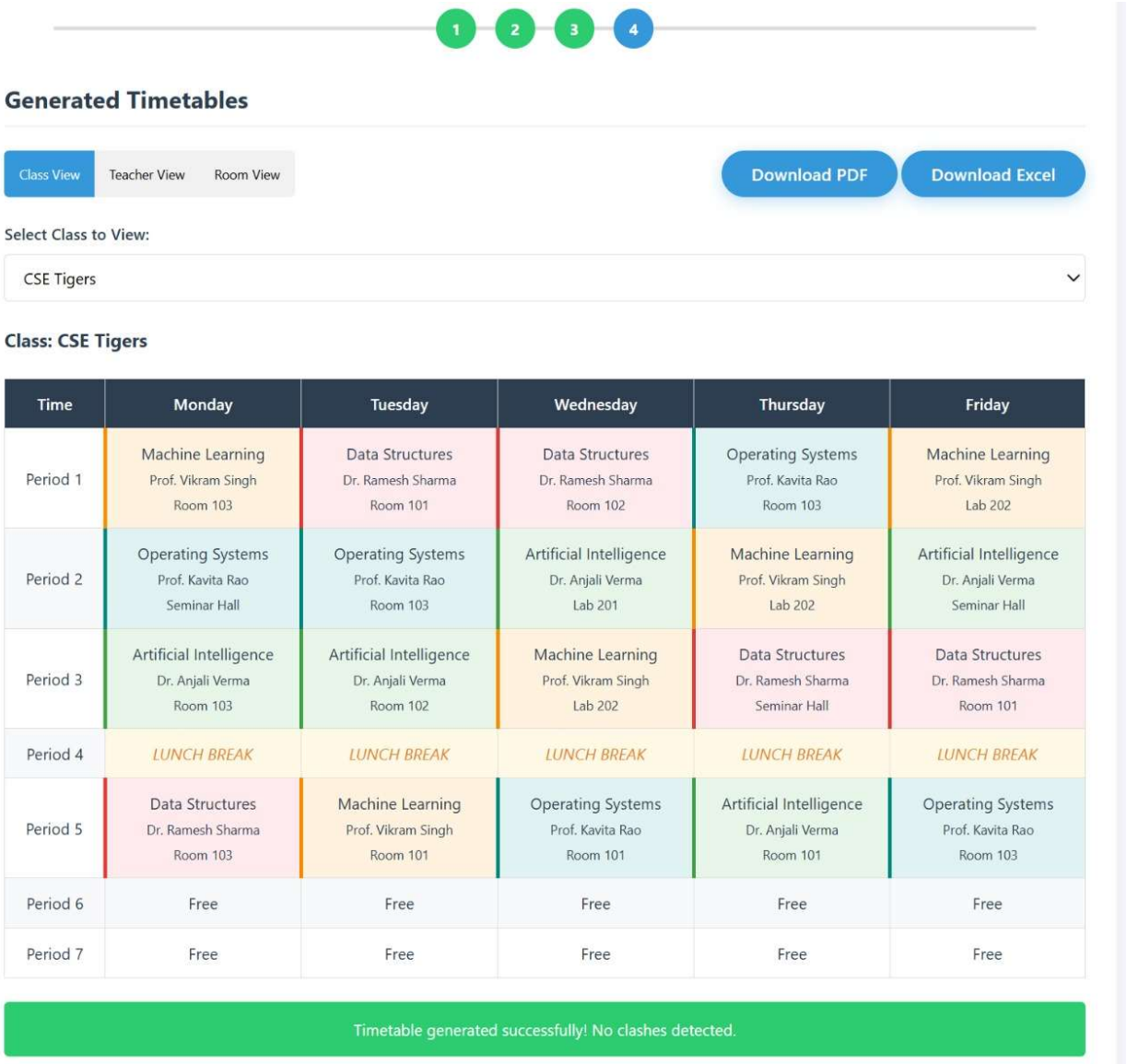
Datasets and Tools

- No external datasets; data is user-input for classes, teachers, subjects, and rooms
- Simulated scheduling scenarios used during testing for validation
- Technologies used: HTML, CSS, JavaScript for front-end and logic
- Libraries: jsPDF (PDF exports), xlsx (Excel exports)
- Google Calendar API for timetable calendar integration
- Testing done with browser developer tools and cross-device compatibility suites

Implementation and Results

Implementation of the project was carried out in several phases following the principles of modular design. The first efforts were directed toward developing an intuitive and responsive user interface for easy input and setup of timetable parameters. Next, the scheduling engine was designed to execute the core allocation algorithm supported by modules controlling conflict detection and user notification. Export facilities were successfully added to produce professional-quality PDFs and Excel sheets. Moreover, the program was thoroughly tested on several desktop and mobile platforms to ensure uniformity of behavior and availability. Visual elements like analytics dashboards were completed to display important measures, such as teacher workload and room usage. The final system consistently generates clash-free timetables alongside meaningful insights and flexible export choices.





Challenges

- Overlapping teacher and room assignments initially caused conflicts; resolved with strengthened validation and user alerts
- Inconsistent responsive design on mobiles; fixed through enhanced CSS modular styling
- Export formatting errors in PDFs and Excel files; mitigated by upgrading and properly configuring export libraries
- Emergency rescheduling concurrency issues (race conditions) fixed by improved state management and asynchronous event handling

- Ensured robustness through iterative testing, thorough bug fixing, and close team collaboration

Applications and Future Enhancement

- Applicable across academic institutions for efficient and clash-free timetable creation
- Reduces administrative workload and improves scheduling accuracy
- Supports multi-format exports and mobile accessibility for diverse user needs

Future improvements:

- AI-driven predictive scheduling adjustments based on historical data
- Integration with campus-wide management systems for centralized administration
- Collaborative multi-user scheduling capabilities
- Machine learning-based personalization and optimization for resources

Individual Reflections

1. **Akkala Harshitha** [Frontend Developer & UI Designer]: My main role was developing the frontend of the AI Timetable Generator web application. I focused on designing the user interface to be clean and intuitive, so that users could easily input classes, teachers, and subjects without confusion. I worked on making the site responsive to work well on different devices like laptops, tablets, and phones. This involved writing HTML, CSS, and JavaScript to create dynamic forms and interactive elements. I also paid attention to accessibility and consistent styling to improve overall user experience. Through this project, I enhanced my skills in frontend web development and learned how important user-centered design is when building functional applications. Testing different screen sizes and browsers helped me understand the complexities of responsive design. It was rewarding to see the interface come together smoothly, making the complex scheduling process simpler for users.

2. **P. Ramya Sai** [Scheduling Algorithm Specialist]: My responsibility was to research, design, and implement the core scheduling algorithm of the project. After studying various scheduling techniques, I chose the graph coloring algorithm to solve the problem of

assigning time slots while avoiding conflicts between teachers, classes, and classrooms. Implementing this method required careful handling of constraints like lunch breaks and maximum periods per day. I coded and optimized the algorithm in JavaScript, testing it against sample data to ensure schedules were generated efficiently without clashes. I also collaborated with the frontend developer to integrate the algorithm output with the user interface. This experience greatly improved my understanding of algorithm design and the practical challenges of constraint satisfaction problems in software engineering. I learned to balance performance and accuracy and developed skills in debugging complex code interacting with many variables.

3. **Anna Mariya Martin** [Testing and Quality Assurance]: My role involved comprehensive testing of the AI Timetable Generator to ensure it worked correctly under various conditions. I developed test cases covering all possible user inputs and interactions, including edge cases like duplicate subjects or conflicting teacher schedules. I tested the software on multiple browsers and devices to confirm compatibility and responsiveness. During this process, I reported bugs related to scheduling conflicts, UI glitches, and export functions. After issues were fixed, I performed regression testing to verify stability. My contributions helped improve the reliability and usability of the system, ensuring that users would have a smooth and error-free experience. Working on this aspect deepened my understanding of software testing processes and reinforced attention to detail in quality assurance.

4. **Hasitha Kocherla** [Documentation and Project Coordination]: As the documentation lead and project coordinator, I was responsible for organizing the development workflow and maintaining clear communication among team members. I drafted this project report, documenting the objectives, technical details, implementation steps, and final outcomes in clear and accessible language. I coordinated meetings to align progress and resolved task overlaps. Additionally, I helped with minor testing and supported frontend and algorithm teams with resources and feedback. This role taught me valuable project management and technical writing skills, including how to present complex technical information logically

for a variety of audiences. Keeping everyone connected and on schedule contributed to the overall success of the project and the production of this complete and cohesive report.

References

1. Chen, M. C., Sze, S. N., Goh, S. L., Sabar, N., & Kendall, G. (2021). *A survey of university course timetabling problem: Perspectives, trends and opportunities*. *IEEE Access, 9*, 106515–106529. https://doi.org/10.1109/ACCESS.2021.3100613
2. Dofadar, D. F., Khan, R. H., Hasan, S., Taj, T. A., Shakil, A., & Majumdar, M. (2022). *A hybrid evolutionary approach to solve university course allocation problem*. *arXiv preprint arXiv:2212.02230*. https://arxiv.org/abs/2212.02230
3. Malviya, A., Agrawal, B., & Kumar, S. (2024). *Graph coloring algorithm for course time table scheduling problem*. *ShodhKosh: Journal of Visual and Performing Arts, 5*(3), 452–460. https://www.granthaalayahpublication.org/Arts-Journal/ShodhKosh/article/view/2182 View of GRAPH COLORING ALGORITHM FOR COURSE TIME TABLE SCHEDULING PROBLEM
4. Mudaliar, V., Rohini, & Mishra, M. K. (2016). *An overview of the heuristic approaches for university course timetabling system*. *International Journal of Engineering Research & Technology (IJERT), 4*(21). https://www.ijert.org/an-overview-of-the-heuristic-approaches-for-university-course-timetabling-system An Overview of the Heuristic Approaches for University Course Timetabling System