Efficient and Automated Approach for Time Table Generation

[1]Tarun G S, [2]Ravi Shivaji Mahipati, [3]V S Krishna Chaitanya Avvari, [4]Shet Reshma Prakash

[1] UG Student SCSE, [2] UG Student SCSE, [3] UG Student SCSE, [4]Assistant Professor SCSE Presidency University, Banglore-560 064

taruntarunyadav2021@gmail.com, ravimahipati011@gmail.com, krishnaavvari2003@gmail.com

reshma.prakash@presidencyuniversity.in

***Abstract*-- Abstract-- As the need for summer term timetabling increases, this project presents a Summer Term Timetable Generation System that is aimed at making the process much more convenient and efficient. Developed with HTML, CSS, JavaScript for the frontend, and Django using SQLite3 for the backend, the system generates academic timetables automatically. It considers vital considerations such as availability of teachers, requirements of the course, how many students are registered, and available room capacities. By so doing, it assists in minimizing human labour and evades some typical issues such as double-booking or clashes of schedules. Data management is handled by Django's backend while SQLite3 offers an easy-to-use, efficient, and light form of storage of schedule details. On the user side, the interface is interactive, clean, and simple to use that can be easily used by administrators as well as faculty members. Smart scheduling algorithms are used by the system which avoid clashes and assign all the resources teachers, rooms, and times fairly. Experiments show that it is effective and can create flexible, flawless timetables. In the future, much scope exists to enhance the system further using AI-based optimization and implementing it on the cloud for increased speed, scalability, and performance.**

1. INTRODUCTION

Manual generation of summer-term timetables is time-consuming and most often leads to errors, scheduling conflicts, and inefficient use of resources. This project solves those issues by providing a web-based system that maximizes the process of generating timetables, thereby making it faster and more accurate. It is built using HTML, CSS, and JavaScript as the frontend technology and Django and SQLite3 as the backend technology. The schedules can be managed by the admins in a simple, interactive manner without any technical knowledge. The smart scheduler considers factors like faculty availability, course constraints, and blocks of time in automatically generating clash-free schedules. Although issues like accommodating varying policies and scaling to big institutions exist, the system is a significant improvement toward smarter, more effective academic scheduling.

2. LITERATURE REVIEW

Oude Vrielink et al. [1] introduced an Auto-Generated Scheduling System (AGSS) to facilitate the timetable to be created more easily in universities. The system employs artificial intelligence to prevent problems such as clashes between rooms and teacher overloading, and was experimented at UiTM using XAMPP and Visual Basic among other tools. [1] R. A. Oude Vrielink, E. A. Jansen, E. W. Hans, and J. van Hillegersberg, "Practices in timetabling in higher education institutions: A systematic review," ResearchGate, Oct. 2017.[Online]. Available: https://www.researchgate.net/publication/320675938

Parkavi et al. [2] conducted a study of timetable generation difficulty and observed that a majority of institutions continue to employ manual scheduling. They considered standard issues and looked at resolution employing algorithms such as Genetic Algorithms, Backtracking, and Local Search methods such as Simulated Annealing and Tabu Search. [2] A. Parkavi, "A Study on Automatic Timetable Generator," M.S. Ramadan Institute of Technology, May 2018. [Online]. Available: https://www.researchgate.net/publication/32626533

Adithya et al. [3] followed the development of timetable algorithms from simple graph techniques to sophisticated ones such as Swarm Intelligence and Genetic Algorithms. They pointed out challenges like system sensitivity and absence of friendly user interfaces.[3] R. Adithya Pai, S. Ashwitha, R. Shetty, and G. Geethalaxmi, "Automated college timetable generator," International Journal of Scientific & Engineering Research, vol. 9, no. 4, Apr. 2018.

Sarthe et al. [4] designed a simple timetable generator that collects course and teacher information to generate conflict-free timetables. Simple as it is, it provides a good foundation for enhancements. [4] M. Sarthe, P. K. Vase, Pradeep, and M. N. R. Mahesh, "Automatic Timetable Generator," IJARCSSE, vol. 7, May 2017. [Online].Available:http://ijarcsse.com/Before_August_2017/docs/papers/Volume_7/5_May2017/SV7I5-0234

3 . PROPOSED SYSTEM

The system facilitates automating and streamlining the production of summer term timetables. It is database-driven to provide accuracy and efficiency.
Frontend: Developed using HTML, CSS, and JavaScript for a responsive and user-friendly interface.

Backend: Implemented with Django for processing and SQLite3 for data storage. Is scalable, stable, and convenient to handle.

Core Modules and Features

1. User Interface Module
• Offers a basic web-based interface for students and instructors.
• Implemented with HTML, CSS, and JavaScript for ease of use and a contemporary look.
• Enables users to search and filter timetables by course, instructor, or time.
2. Input Processing Module
• Gathers required inputs such as course information, instructor availability, and time preference.
• Verifies data to prevent errors or inconsistencies.
• Detects and prevents conflicting or overlapping schedules.
3. Timetable Generation Module
• Utilizes Django to generate optimized timetables in real-time.
• Automatically adjusts to changes in faculty availability or course loads.
• Provides for efficient use of time slots, faculty, and classrooms without conflicts.
4. Database Management Module
• Stores all course, faculty, and scheduling data utilizing SQLite3.
• Provides rapid access, easy updates, and dependable data management.
• Allows administrators to make easy changes or updates to schedules.
5. Output Module
• Presents the final timetable in a clean and readable format.
• Users can download or print their schedules.
• Updates are displayed in real-time so that everyone always sees the most recent version.
6. Error Handling and Logging Module
• Automatically detects and corrects scheduling errors.
• Logs any system or database problems for future troubleshooting.
• Treats invalid inputs gracefully, keeping the system stable and user-friendly.

4. ALGORITHM

```
Step 1: Entry (Home1)
- Display: About Us, Help, Contact Us, Login
- On the basis of selection, display info or proceed to Login
Step 2: Login
- Input username/password
- If correct → Proceed to Home2
- If incorrect → Display error, remain on login page
Step 3: Registration
- Get user info, validate, store
- Display success message
- Redirect to Home2

Step 4: Home2 Options
- Insert: Teacher, Room, Timing, Course, Department, Section
- Generate Timetable
Step 5: Data Entry
- Receive input, validate, save to database
Step 6: Generate Timetable
- Retrieve all data
- Enforce rules
- Save timetable
Step 7: Logout
- Terminate session
- Go to login page => Home1
Step 8: End
```

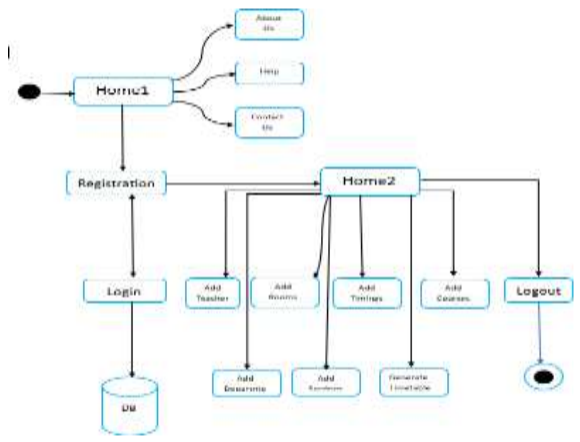

Figure 1. Architecture of Timetable Generation Application

The above diagram shows a system workflow of the timetable management. The modular design and navigation between the different components are easily depicted. Below is the detailed description:
1. Home Page (Home1)
About Us: Provides information about what the system does and why it's useful. Help Center: Offers easy, step-by-step instructions to assist users in getting the most from the system. Contact Support: If users require help or encounter problems, they can quickly contact support.
2. Signup and Login Signup:
New users can easily sign up to access all the features of the system. Login: Registered users can securely login to view their customized information. All the information is securely stored in a central database.
3. Dashboard (Home2) After logging in, users are taken to the dashboard where They can: Add Teacher: Add details of the teachers for the timetable. Add Rooms: Add information about the available classrooms. Add Timings: Add available time slots for classes or events.
4. Key Features
Course Management: Quickly add and administer courses, names, codes, and other details. Services Management: Maintain and organize department-related data. Service Integration: Include any additional services required for optimal scheduling.
5. Timetable Generation & Database
Automatically generates a conflict-free, optimized timetable from what the users input. Logout: Log out securely when finished. Central Database: Everything of significance (courses, rooms, timetables, etc.) is kept in one location for convenience.
6. User Experience Easy Navigation: The system is made to be easy and user-friendly, so users can navigate from login to scheduling with ease. Efficient Scheduling: It automatically allocates teachers, rooms,

and time slots in a manner that optimizes usage of available resources without conflicts.

7. Security & Sessions

The system provides safe login and protects user information. Session management provides a smooth ride and ensures user information remains intact during their stay on the site.

5. KEY FEATURES

1. Painless Course and Faculty Management The system is designed based on Django, which makes all the critical information such as courses, faculty availability, and student registrations easy to manage. It also employs a minimal and portable database known as SQLite3 to store and retrieve this information with ease [3].
2. Simple and Accessible Web Interface The user interface is achieved through the assistance of html, css, and Java script to allow for ease of use. It presents a clean and responsive look, therefore offering an easy means of administrators to manage timetables.
3. Simplified Timetable Generation The computer produces timetables automatically based on What staff members are on hand, which lessons need to be allocated, and what the time slots are. It eliminates wastage of time, clears out errors, and optimizes available resources.
4. Smart Scheduling with Rules It works under smart rules such that problems are not faced—such as a classroom not being scheduled twice, or a teacher not being scheduled with conflicting classes. It also considers hierarchies between similar classes such that all functions harmoniously.
5. Ready to Grow and Change Anytime Due to the solid Django backend, the system can grow with the institution's expanding needs. It also accommodates live changes, allowing administrators to alter or update
the schedule in a rush when it is needed best for busy or large institutions.

6. MATHEMATICAL MODEL

The process of designing a summer term schedule involves four very important steps that together form a schedule that's efficient, accurate, and manageable. The system uses Django and SQLite3 in the backend and HTML, CSS, and JavaScript on the frontend to make sure everything runs perfectly.

1. Entry and Validation of Data

Users start with inputting such information as what teachers are on duty, which classes are running, and which classrooms are vacant. The system cross-checks this information exhaustively to ensure that nothing is omitted or out of place. If there's an issue, it alerts the user before moving on.

2. Auto-construction of the Schedule

After all the information is guaranteed to be correct, the system will automate and automatically assign teachers, classes, and rooms. The system also has some rules that it employs to avoid conflict between classes and to maximize the usage of time and space[4].

3. Managing Conflicts

If there is any issue that comes up—e.g., having a teacher assigned in two different classes at the same time—the system displays as such. It may display the other options or give an admin override to make a decision based on whatever is prioritized the most.

4. Making the Timetable Better

Lastly, the system checks the schedule for balance. It ensures not to overbook any teacher's schedule, and students' times do not conflict, and so on, in accordance with school codes. It is also flexible in that the time can be done at any desired time.

7. ADVANTAGES

1. Easy to Use and Manage The system simplifies course management, instructors, and classrooms, and the administrators are able to handle everything easily in one place.
2. Saves Time and Reduces Mistakes It reduces effort usage, removes scheduling errors, and gives proper resource utilization.
3. Simple and Interactive Interface Since it was developed using HTML, CSS, and JavaScript, the user interface is simple, interactive, and updates instantly—easy and quick scheduling.
4. Adaptable and Built to Grow With the backend handled by Django and SQLite3 as the database, the system can easily grow with larger institutions and is also very adaptable for future changes.

8. CHALLENGES OF THE SYSTEM

1. Dependence on Algorithm

How intelligent and effective the schedule algorithm is will to a large extent determine whether the system can produce good timetables. If poorly optimized, you may end up with the same class scheduled twice, room double-booking, or teacher clashes.

2. Handling Last-Minute Changes It is not an easy task for the system to shift gears so quickly and accommodate last-minute adjustments—like a substitute teacher calling in sick, an unexpected shift in course enrollments, or finding that a room is unavailable. To update this and not put the rest of the schedule on its head is a tough question.
3. Performance Under Pressure Django is a great framework, but when the system is operating over a large amount of data or highly complicated timetables, it is sluggish—particularly if the database queries are not properly optimized.
4. SQLite3 Limitations SQLite3 is okay for small programs, but it won't be capable of handling if there is plenty of data, plenty of users at the same time, or complex operations. In these cases, it will slow down the process or introduce delays in retrieving information.
5. Keeping Everything Working Together The system integrates different tools—ranging from the front end (HTML, CSS, JavaScript) to the back end (Django and SQLite3). Getting them to work together in harmony is difficult, and if one of them is changed, it might affect everything else. This makes it difficult to update and maintain.

9 . CONCLUSION

The Summer Term Timetable Generation System provides a contemporary and effective solution to academic timetabling by fully automating the process with Django , sqlite3, html, css, and JavaScript. It minimizes manual labour, prevents scheduling conflicts, and optimizes the utilization of institutional resources. The friendly interface and real-time updating features of the system allow administrators to manage timetables efficiently.

Although created at first for planning during the summer term, it can be employed during the whole academic year because of its modularity. It is also scalable and economical to implement in any-sized institutions. Upcoming updates such as conflict detection based on AI, support for mobile, and integration into the cloud will further enhance its flexibility and functionality. As online education continues to become more widespread, such systems will become even more critical to automating scholarly activities and increasing overall efficiency.

10. REFERENCES

[1] R. A. Oude Vrielink, E. A. Jansen, E. W. Hans, and J. van Hillegersberg, "Practices in timetabling in higher education institutions: A systematic review," ResearchGate, Oct. 2017. [Online]. Available: https://www.researchgate.net/publication/320675938
[2] Parkavi A., "A Study on Automatic Timetable Generator," M.S. Ramadan Institute of Technology, May 2018. [Online]. Available: https://www.researchgate.net/publication/32626533
[3]. Pillay, N., & Qu, R. (2022). An Evolutionary Algorithm for the Multi-Criteria University Timetabling Problem. Applied Soft Computing, Volume 115, Article 108163. [https://doi.org/10.1016/j.asoc.2021.108163]
[4] Adithya R. Pai, Ashwitha S., Raksha Shetty, and Prof. Geethalaxmi, "Automated college timetable generator," International Journal of Scientific & Engineering Research, vol. 9, no. 4, Apr. 2018.
[5] M. Sarthe, P. K. Vase, Pradeep, and M. N. R. Mahesh, "Automatic Timetable Generator," IJARCSSE, vol. 7, May 2017. [Online]. Available: http://ijarcsse.com/Before_August_2017/docs/papers/Volume_7/5_May2017/SV7I5-0234
[6] M. S. Shaaban et al., "Recent Advances in Automatic Timetable Generation: A Comprehensive Review," 2017.
[7]. Wang, X., & Xu, H. (2021). A Novel Memetic Algorithm for Solving University Timetabling Problems. Expert Systems with Applications, Volume 178, Article 115018. [https://doi.org/10.1016/j.eswa.2021.115018]
[8]. Nguyen, T. T., & Le, M. T. (2021). A Deep Reinforcement Learning-Based Approach for Automated Course Scheduling. IEEE Access, Volume 9, 2021, Pages 115765-115778. [https://doi.org/10.1109/ACCESS.2021.3106042]
[9]. Oladipo, W. K., Bamidele, A. O., & Olalekan, A. M. (2019). Automatic Timetable Generation using Genetic Algorithm. International Journal of Applied Information Systems, Volume 12, Issue 19, Pages 1–3. [https://www.ijais.org/archives/volume12/number19/1047-2019451779]
[10]. Kralev, V., Kraleva, R., & Yurukov, B. (2016). An Event Grouping Based Algorithm for University Course Timetabling Problem. arXiv preprint, arXiv:1607.05601. [https://arxiv.org/abs/1607.05601].