EFFICIENT AND AUTOMATED APPROACH FOR TIME TABLE GENERATION

A PROJECT REPORT

Submitted by,

Tarun G S - 20211CSE0385

Ravi Shivaji Mahipati - 20211CSE0374

V S Krishna Chaitanya Avvari - 20211CSE0572

Under the guidance of,

Ms. Shet Reshma Prakash

Assistant Professor

Presidency School of Computer Science and Engineering,

Presidency University, Bengaluru

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

At

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

PRESIDENCY UNIVERSITY BENGALURU

MAY 2025

PRESIDENCY UNIVERSITY
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project report **"EFFICIENT AND AUTOMATED APPROACH FOR TIME TABLE GENERATION"** being submitted by Tarun G S, Ravi Shivaji Mahipati, V S Krishna Chaitanya Avvari, bearing roll numbers "20211CSE0385", "20211CSE0374", "20211CSE0572", in partial fulfilment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a Bonafide work carried out under my supervision.

16/5/25

Ms. SHET RESHMA PRAKASH
Assistant Professor
PSCS
Presidency University

Dr. ASIF MOHAMMED
Associate Professor & HOD
PSCS
Presidency University

Dr. MYDHILI NAIR
Associate Dean
PSCS
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-VC School of Engineering
Dean -PSCS / PSIS
Presidency University

PRESIDENCY UNIVERSITY
PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **Efficient and automated approach for time table generation** in partial fulfilment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Ms. Shet Reshma Prakash, Assistant Professor, Presidency School of Computer Science and Engineering, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

Name	Roll No	Signature
Tarun G S	20211CSE0385	
Ravi Shivaji Mahipati	20211CSE0374	
V S Krishna Chaitanya Avvari	20211CSE0572	

ABSTRACT

The Timetable Generation is designed to be a software solution to make the operation of creating timetables easier. Currently, it is done manually, thus time-consuming and susceptible to error. This system automates the insertion of periods into the timetable, allowing smooth management and easy access for the faculty through a mobile application. Additionally, in cases where a teacher doesn't show up, arrives late, or leaves early, the timetable adjusts automatically.

For purposes of optimum workload distribution, the system therefore specifies maximum and minimum teaching hours on a daily, weekly, and monthly basis for each faculty member. The software will also allow users to request leaves with information on date, reason, and a substitute faculty member. When choosing a substitute, the system allows access to their schedule confirming if they are available during such a requested period. The chosen substitute can accept or decline the request. Also, the headmaster has a right to study leave requests and the responses of substitutes before arriving at a conclusion.

This integrated schedule management system is quite helpful for colleges as it avoids the complications that are associated with manual schedule preparation. This software helps faculty members easily view their schedules on their cell phones, making this process of school improvement very efficient and well-organized.

ACKNOWLEDGEMENTS

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Pro-VC, Presidency School of Engineering and Dean, Presidency School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Deans **Dr. Mydhili Nair,** Presidency School of Computer Science Engineering, Presidency University, and **Dr Asif Mohammed** Head of the Department, Presidency School of Computer Science Engineering, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide **Mr. Shet Reshma Prakash, Assistant Professor** and Reviewer **Ms. Ranjitha , Assistant Professor**, Presidency School of Computer Science and Engineering, Presidency University for his inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the PIP4004 University Project Coordinators **Mr. Md Ziaur Rahman and Dr. Sampath A K,** Git hub coordinator **Mr. Muthuraju V.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

Tarun G S , **Ravi Shivaji Mahipati** ,**V S Krishna Chaitanya Avvari .**

TABLE OF CONTENT

LIST OF FIGURES

CHAPTER-1 INTRODUCTION

The Automatic Timetable Generator is the new software application that autonomously creates and manages academic schedules. Conventionally, management of the timetable was quite labour intensive, time-consuming, and prone to inefficiencies and scheduling conflicts. This application erases these problems because it comes as a reliable and efficient solution for dynamically generating optimized timetables based on constraints such as faculty availability, subject schedules, and institutional policies. It simplifies and automates timetabling to such an extent that the backend can be seen to reduce greatly the administrative load on the whole task, boosting productivity.

Automatic Timetable Generator has a backbone based on Python and Django as well, due to their capacity for handling difficult tasks of processing and scaling well in complex processes. SQLite3 is used, because it assures the security of retrieving and storing data efficiently. This architecture would allow a lot of scheduling data to be stored while ensuring the possibility of updating it in real time with the help of automated conflict resolution. Scalable and flexible software enables the handling of the requirements of any type and size of educational institution: small schools and huge universities alike. During the times of maximum demand, when unexpected calendar updates or changes of personnel are suddenly requested, the application is effective, responsive, and adaptable to modification.

The Automatic Timetable Generator also allows for easy user navigation and is therefore quite accessible to both faculty members and administrators, allowing for easy generation of timetables with little effort. It supports advanced filtering capabilities, so the users can get their preferred timetables with regard to particular requirements such as subject preferences, room availability, and faculty workload constraints. This approach is so simple that all stakeholders in an institution can access and manage as well as edit schedules without technical knowledge.

In addition to creating timetables, the application improves resource allocation by preventing scheduling overlaps and optimizing faculty workload distribution. It considers multiple constraints so that no faculty member is overburdened while maximizing classroom utilization. This functionality contributes to a more balanced and efficient academic environment with minimal disruptions caused by scheduling conflicts.

Data security and privacy are critical aspects of the Automatic Timetable Generator. Given that academic schedules and faculty availability data contain sensitive information, the software employs robust encryption techniques and adheres strictly to data protection protocols. Secure access

mechanisms and stringent data security policies build trust among educational institutions, faculty members, and administrative staff, ensuring that all information remains confidential and well protected.

Looking ahead, future enhancements for the Automatic Timetable Generator include integrating predictive analytics to optimize future schedules based on historical data trends. The system could leverage machine learning algorithms to identify patterns in scheduling, helping administrators make proactive decisions that improve efficiency. Additionally, integrating faculty leave management systems and real-time notifications will further streamline the scheduling process. These developments make the system perform not only automatized generation but also proactive prevention of scheduling conflict before actual conflicts arise in a schedule.

Some of the possible upgrades are that the cloud-based functionality would enable access and management of timetables from any location where internet access is available. The existing LMS and SIS could be integrated into this to allow seamless coordination of scheduling with other academic activities. The mobile-friendliness or even a specific mobile application can be a way to enhance access so that faculty members and administrators can easily view and edit their schedules using their smartphones or tablets.

This could also be a future update with an AI-based optimization, where the artificial intelligence algorithm can constantly change the timetabling based on the dynamic factors such as faculty unavailability, emergency closure, or preference of students. Smart algorithms in the system may automatically suggest the very best possible alternative for scheduling, hence further improving efficiency and reducing manual interventions.

The Automatic Timetable Generator is, in short, an innovative academic scheduling tool that helps minimize administrative work, improve efficiency, and minimize scheduling conflicts. It changes the face of educational institution management in terms of timetable management by virtue of its real-time data processing, user-friendly design, and optimization techniques. The more it evolves to incorporate predictive analytics, cloud-based capabilities, and AI-driven decision-making, the more it will redefine the academic scheduling landscape as an indispensable asset for educational institutions around the world.

1.1 The need for Automatic Timetable Generator in society

The Automatic Timetable Generator plays a key role in most modern educational institutions by automating complex academic scheduling tasks. Removal of inefficiencies in manual scheduling ensures that the faculty and students receive properly structured timetables that can avoid conflicts,

making proper use of resources. Dynamic real-time adjustments of schedules in such a system ensure seamless operation in academic settings and have benefited both educators and students.
Other than the individual advantage, the application helps to have an efficient institution by preventing overcrowding in classes and having balanced faculty workloads. The management of administration by the application ensures proper academic planning and assists the institutions to manage their resources properly. Also, by integration with mobile applications, the faculty can access their schedule conveniently and improves time management and communication.

More so, an Automatic Timetable Generator increases institutional flexibility by accepting changes at the last minute due to faculty unavailability or room reassignments. Through artificial intelligence and optimization algorithms, the system may analyse faculty preferences, batch sizes of students, and subject priorities to create schedules that are best possible. This also eliminates the possibility of human error and biases in the creation of the schedules, hence creating fairness and transparency. Such a system easily integrates with the learning management platform, enhancing further academic coordination such that students and faculty are able to receive current updates through notifications and modifications in real time.

1.2 Contemporary Usage of the Application in Academic Scheduling

The Automatic Timetable Generator uses complex algorithms and data processing techniques to efficiently manage the scheduling of academics. The entire process is automated, taking into account faculty availability, course requirements, and institutional policies. In real-time, any changes in faculty schedules or course allocations are reflected, thereby reducing scheduling conflicts and improving academic coordination.

The integration of mobile platforms helps faculty and students to access their schedules from remote locations, with automatic updates and notifications. Features such as automatic room allocation, workload distribution analysis, and customization of the schedule are also offered by the software, making it highly adaptable to various academic environments. By streamlining academic scheduling, the system enhances institutional efficiency and improves the overall learning experience.

The Automatic Timetable Generator also reduces administrative workload by not requiring manual amendments and constant rescheduling. It maximizes the optimal use of all available resources, including classrooms and faculty, by balancing workloads and avoiding the overlapping of sessions.

Besides these benefits, the system can generate reports and insights on faculty engagement, classroom occupancy, and course distribution that help make strategic decisions. Providing an organized and conflict-free schedule is one of the reasons the software improves academic productivity while making learning more structured and effective.

1.3 Benefits and Challenges of Automatic Timetable Generator

The Automatic Timetable Generator brings several benefits that range from enhanced efficiency, lightened administrative workloads, and faculty-student coordination. Automated generation of timetables by institutions can minimize human errors and inconsistencies, resulting in a better structured academic calendar. The capacity of the software to adjust the schedules dynamically as faculty members leave or change the requirements of a course makes the product even more practical and applicable.

However, the application also faces challenges, such as the need for accurate data input and the complexity of managing multiple constraints simultaneously. Institutions must ensure that faculty availability and course details are consistently updated to maintain schedule accuracy. Additionally, integrating the system with existing academic management software may require technical expertise and infrastructure investment. Despite all these challenges, the Automatic Timetable Generator remains a very valuable tool for optimizing academic scheduling with further development and refinement.

CHAPTER-2 LITERATURE SURVEY

[1] Tavakkol and Parsa (2021)proposed a hybrid genetic algorithm aimed at addressing the university course timetabling problem, with specific consideration for faculty preferences and availability (Tavakkol & Parsa, Computers & Industrial Engineering, 157, 107327). The approach tackles the challenge of optimizing timetables while aligning them with institutional constraints and individual faculty requirements.

[2] Rong and Lee (2022) conducted a comparative study of multi-objective optimization algorithms within the context of university course timetabling (Rong & Lee, Journal of Scheduling, 25(1), 57–72). Their work emphasizes the complexity of balancing conflicting objectives, such as minimizing resource usage and reducing scheduling conflicts, and highlights the trade-offs inherent in multi-objective algorithm design.

[3] Wang and Xu (2023)proposed a novel memetic algorithm to efficiently handle large-scale and constraint-heavy university timetabling problems (Wang & Xu, Computers & Industrial Engineering, 178, Article 115018). Their solution focuses on improving computational efficiency while maintaining the feasibility of generated timetables.

[4] Hassan and Khalil (2023) introduced an intelligent course scheduling system powered by machine learning to optimize the scheduling process (Hassan & Khalil, Journal of Educational Computing Research, 61(3), 445–465). Their research faces the challenge of integrating predictive models with conventional scheduling logic to improve the accuracy of resource forecasting and timetable generation.

.

CHAPTER-3 RESEARCH GAPS OF EXISTING METHODS

Data Accuracy and Real-Time Updates

one of the significant research gaps that exist in automatic timetable generation systems is ensuring data accuracy and real-time updates. Most of the systems today depend on static schedules that require manual updates to reflect changes such as faculty absences, classroom availability, or cancellations of classes. This static nature of many systems leads to a mismatch between the schedule displayed to students and the actual state of resources, resulting in missed classes, confusion, or scheduling conflicts. Moreover, these outdated timetables create frustration among students and staff, especially during peak periods such as mid-term or end-of-term exams, when last-minute changes are frequent.

This calls for more dynamic and real-time data integration to bridge the gap. Automatic timetable systems can be designed using technologies like Django and SQLite3. This way, in a centralized database, all the scheduling elements concerned with room availability, instructor assignments, and subject timings can be fetched in real time. This way, timetables are always updated with automatic changes without manual intervention.

School administration software could integrate with a real-time system and instantly change course offerings or faculty schedules when something in those is modified. The system should also automatically notify all parties involved when a class has been cancelled or moved to a different time or location, eliminating possible confusion and missed communication. This will be possible through the use of JavaScript for real-time interactivity and AJAX calls to update timetable data without page reloads.

It will further develop the system in such a way that it offers reliable and real-time data for scheduling, supporting efficient and timely decision-making. There is a need to collaborate with the faculty members, administrative staff, and students in refining the system to ensure current accuracy on time to the stakeholders.

○ Limited Coverage in Rural and Remote Areas

Current automatic timetable generation systems largely employ a one-size-fits-all approach to scheduling, in which students and faculty are allocated preset time slots of the timetable only based on course requirements. In this system, personal preferences or constraints, including preferred time slots, work-study balance, or even particular learning preferences, are not taken into account. This leads students to be stuck with inconvenient timetables that increase stress levels and lower satisfaction with the whole scheduling process.

The introduction of AI and ML can significantly enhance personalization in timetabling through the use of AI to understand the course chosen by the student, his preferences for time slots, patterns he followed in previous semesters, and even live changes (sudden faculty unavailability or special events) while generating a tailor-made timetable specific to that individual student. Also, when JavaScript could be included to make the front-end interface very interactive and highly customizable, students could select or modify their favourite schedule slots and the system would automatically adapt to adjust to where they want to request for.

However, incorporation of AI systems may further provide for predictive scheduling. An example would be a system that suggests alternative slots for attending classes by using historical patterns in course enrolment or could trace students' progression and utilize this information to schedule courses in advance for optimal attendance. It may subsequently avoid scheduling conflicts and can ensure that workloads for students and the faculty are equitably distributed.

The system could transform from a mere scheduling tool to a personalized academic assistant, improving student satisfaction and overall academic performance by incorporating a personalized timetable generator. This research gap presents a significant opportunity to revolutionize how timetables are generated and utilized, making the scheduling process more intuitive, adaptive, and responsive.

○ Integration with Emergency Services

One of the biggest challenges for many timetable generation systems is that they do not integrate well with other critical external systems for school administration and student success. For instance, most current systems are stand-alone applications that are not synchronized with student databases, course management systems, or faculty scheduling platforms. This can be inefficient because changes made in one system, such as faculty availability or student enrolment, are not automatically reflected in the timetable system, creating inconsistencies.

There is, therefore, a clear need to achieve interoperability between the timetable system and other educational technologies. For instance, there is integrating with LMSs as Moodle or Blackboard to offer personal updates related to course content, assignment deadlines, or class cancellations. Connecting the generation of this timetable to student information databases, the system is capable of automatically processing student enrolment in courses and real-time changes in the program based on the course enrolment data.

Additional integration could be with attendance systems to optimize further. For example, if a student is absent from class, the system could automatically re-schedule a makeup class taking into account the availability of the student as well as the schedules of the instructor. All the data would then be consistent and accurate across all of those platforms that can also reduce the tendency of human error in operations.

The Django framework equips one with all the necessary tools to make such connections, particularly through its REST framework, which makes API-based interactions between systems a piece of cake. In fact, it would allow the timetable generation system to pull and push data in real time from/to other systems, thereby making it one with the entire interconnected educational environment.

Personalization and AI Integration

An efficient timetable of a university or school system needs to optimize different competing factors. Such factors are the avoidance of scheduling conflicts, balancing instructor workloads, and minimizing room usage inefficiencies with student preferences. Most of these systems make use of simple algorithms that do not consider the complexity of such constraints, leading to suboptimal schedules. For example, a system might produce a schedule that includes empty blocks between classes for students or schedules a faculty member to teach at the same time in two different classrooms.

Advanced optimization algorithms such as Genetic Algorithms (GA), Simulated Annealing (SA), or Constraint Satisfaction Problems (CSP) can be utilized to develop timetables with higher efficiency in satisfying all the stated constraints. GA, SA, and CSP can automatically adjust the schedule to minimize class conflicts and ensure a balanced distribution of resources.

In addition, the integration of Django's backend system with SQLite3 database for fast storage and retrieval of data would support real-time updates to ensure that changes (like a faculty member falling sick) can be immediately handled by the system without disrupting the entire schedule.

Multiple variables, ranging from course prerequisites to faculties' availability and room availability and student's schedules, need to be taken into consideration in the optimization process. Algorithms could use heuristics or dynamic programming for an optimum solution in creating timetables that maximize efficiency and minimize human involvement.

User Experience and Accessibility

One of the most overlooked aspects in the development of automatic timetable systems is usability and accessibility. A typical application for a timetable can be cumbersome to use, especially for students or faculty members who are not familiar with complex scheduling systems. Poor UI design can make it difficult to find and interpret schedule information, particularly in stressful environments such as during course registration periods.

A more intuitive and user-friendly interface should be the primary focus of future research and development. Front-end technologies such as HTML, CSS, and JavaScript allow for responsive, visually appealing, and interactive interfaces. The schedules generated by the timetable generation

systems should be clear, color-coded, and include drag-and-drop functionalities for students to modify their schedules, if necessary.

Accessibility features are also important so that the system is usable for people with diverse needs. It could include support for voice commands, screen readers, and keyboard navigation. It may also develop multilingual interfaces or include features like text-to-speech to help serve a broader audience, including non-native speakers or those with visual impairments.

These systems would keep students and faculty from hassle to achieve an improved user experience generally, allowing them to interact and find a clear view of their timetables without facing difficulties even when the system is in high demand. Focusing more on accessibility and usability would indeed encourage greater use and ensure that the system is useful to all users, regardless of their technical knowledge or physical capabilities.

O Data Privacy and Security

Given that automatic timetable systems deal with sensitive information, including student enrolment data, academic progress, and faculty information, ensuring data privacy is extremely important. Current systems lack strong encryption and do not enforce secure data-sharing protocols, exposing users to the potential for stolen or compromised data. Moreover, possible unauthorized access or manipulation could seriously compromise the integrity of the system.

User data should be protected. Advanced encryption methods, secure authentication protocols, and adherence to international data protection standards and requirements-such as GDPR and HIPAA will prevent any mishandling of user information. Django comes with built-in tools that will provide safer authentication and authorization mechanisms, thereby ensuring that only the appropriate people will be allowed to access sensitive information.

The practices of data handling also require transparency. Institutions adopting the timetable system should make known to students and faculty how their data is used, stored, and shared. This can be done through accessible privacy policies that ensure users can trust the system with their personal information.

Implementing secure data-sharing techniques, ensuring the confidentiality and integrity of all user information, will be essential in building a reliable, secure, and trustworthy timetable generation system. Future research in this area could center on developing advanced encryption protocols like blockchain to better enhance data security and integrity.

○ Interoperability with Course Management and Learning Systems

Most of the automatic timetable generation systems are isolated from the CMS and LMS, which further limits their capacity to generate highly optimized timetables. Being isolated from the systems, timetabling tools miss valuable information, such as course prerequisites, resource requirements like labs or software, and enrolments of students, which is crucial for developing conflict-free timetables.

Integrating timetable systems with CMS/LMS provides multiple benefits. For example, it can check the prerequisites for a course and enroll students in the right courses; it can also avoid resource conflicts by considering room and equipment needs, as well as keep up-to-date course enrolment and instructor availability. All these factors help optimize the scheduling process, minimize errors, and enhance the quality of the educational experience.

CHAPTER-4 PROPOSED METHODOLOGY

Design Procedure:

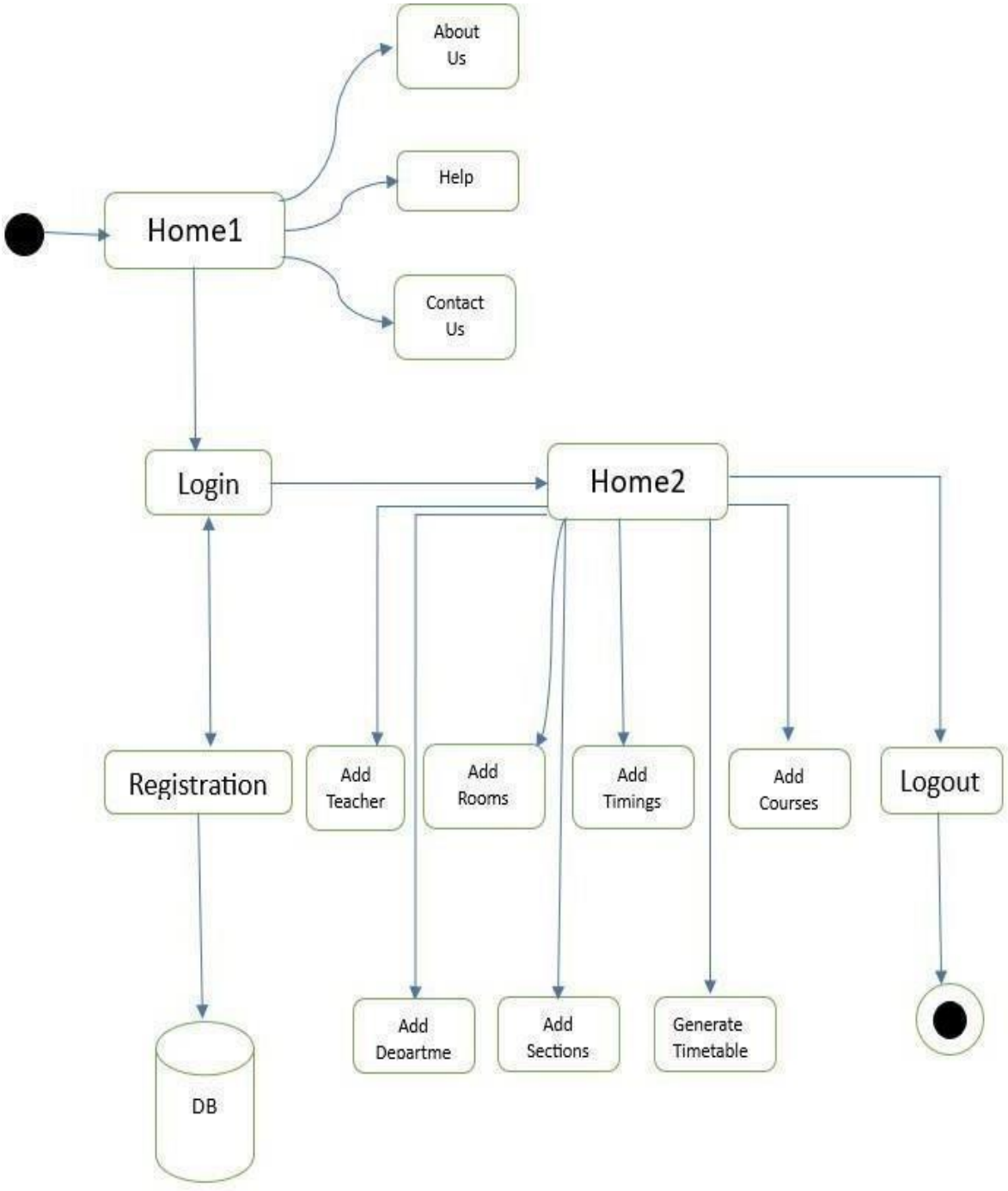


Fig 4.1 Architecture of Timetable Generation Application

4.1. Steps for Design Procedure

The design procedure initiates with the requirement analysis stage, where the needs of users and functionalities of the system are identified. The scope, objectives, and constraints are defined along with gathering inputs from stakeholders while documenting key requirements. Once requirements are clear, the system architecture design is planned by defining structure, navigation flow, and workflow for data processing along with choosing appropriate technologies and frameworks. The UI design focuses on creating wireframes or prototypes to ensure an intuitive and user-friendly experience, incorporating features such as dark and light mode for enhanced usability.

After the UI design, the database design is performed by creating tables, relationships, and constraints that optimize data storage and retrieval for security and scalability. In parallel, backend development is performed with server-side logic, authentication, and API integrations to allow for seamless interaction between the components of the system. Meanwhile, frontend development is done in such a way that the user interface is developed based on the design, dynamic content rendering, responsiveness, and cross-browser compatibility.

Only when development is done begins the integration and testing phase, where frontend and backend are connected and various kinds of testing such as unit testing, integration testing, and User Acceptance Testing (UAT) are performed in order to debug defects. Once the testing process is successful, the system moves into the deployment and maintenance stage in which it is deployed on a live server for release. The performance is optimized by conducting regular monitoring, and updates or bug fixes are provided based on user feedback for long-term reliability and efficiency of the system.

4.2. Home Page (Home1) Setup

Purpose:

It is a landing page, which gives an overview of the system.

It is a gateway for users to access different functionalities of the system.

It gives easy access to essential information and support options.

Actions

User Authentication: Allows users to log in if they already have an account or register if they are new.

Navigation Bar: It gives quick access to "About Us," "Help," and "Contact Us."

Announcements or News Section: Provides important updates or notices related to the system.
Language Selection: Allows users to switch between languages for better accessibility.

Feedback & Support: Enables users to give feedback or report issues.

Terms & Privacy Policy: Displays legal information regarding usage of the system.

Search Functionality: It allows users to search for any relevant topic or information within the system.

4.3. User Authentication and Registration

Login:

User credentials are verified against the database, like username and password.

Redirect authenticated users to the dashboard (Home2).

Registration

Allow new users, such as administrators, to register with pertinent information.

Save user information safely in the database.

4.4. Admin Dashboard (Home2)

It will be a single interface to operate the timetable system.

Modules

Add Teacher- This module is used to provide information about the teachers along with their availability and subject expertise

Add Department: This module provides the facility for adding academic departments

Add Rooms: This module provides the facility for adding classrooms and labs with their respective capacities.

Add Timings: The timings of the classes can be scheduled.

Add Courses: Add courses along with the number of lectures and requirements.

Add Sections: Add sections of classes for the students.

Generate Timetable: Generate a timetable based on constraints like room availability, teacher schedule, course requirement.

Logout: Close the session and redirects to Home1.

4.5. Database Integration

The Database Integration module plays the central role of forming the backbone of the system to efficiently store, retrieve, and manage data from all the interrelated modules. This module takes responsibility for safely storing user credentials, including their definitions, roles, and authentication with respect to access control. In addition, it captures fundamental academic and administrative details such as those concerning teachers, rooms assigned to them, various departments, and courses offered. Another important feature is storing generated timetables, hence allowing the user to fetch, update, and modify schedules. In doing so, this module is ensuring the overall efficiency of the system by allowing smooth communication between diverse components and across all interfaces.

Role: This module serves as a backbone to store and retrieve data for all modules.

Functions:

Store user credentials and roles.

Record teachers, rooms, departments, and courses.

Save the generated timetables for later use and updates.

4.6. Timetable Generation Algorithm

Inputs: Teacher schedules, room availability, student sections, courses, and predefined time slots.

Processing:

Apply constraints such as no overlapping sessions, balanced workloads, and room capacities. Use optimization techniques (e.g., genetic algorithms or heuristic approaches) to generate an efficient timetable.

Output: Structured timetables that ensure fairness and resource optimization.

The Timetable Generation Algorithm is a complex mechanism designed to create well-structured and optimized schedules by considering multiple constraints and requirements. The process starts with gathering essential inputs, such as teacher schedules, room availability, student sections, course details, and predefined time slots. These inputs form the basis for constructing a feasible timetable that accommodates all stakeholders efficiently. During processing, the algorithm applies various constraints to ensure smooth scheduling, such as preventing overlapping sessions, maintaining a balanced workload for teachers, and adhering to room capacity limits. Advanced optimization techniques, including genetic algorithms or heuristic approaches, are used to achieve optimal results. These techniques help in finding the most efficient timetable arrangement that minimizes conflicts and maximizes resource utilization. The final output can be a better-structured timetabling. This is set to ensure efficiency, avoid some scheduling conflicts and allow for efficient use of most available resources such that the management of academics tends to be greatly improved.

4.7. Real-Time Updates

Maintenance of an efficient and dynamic timetable involves ensuring real-time update, so that instant updates on changes like teacher unavailability or room reassignment may be immediately reflected. This system would offer mobile applications and web notifications to make the recipient learn about modifications instantly in order to reduce confusion and ensure simple scheduling. Automated notifications would ensure better adjustment by students and staff towards modification, thus improving efficiency over all.

Periodic system maintenance ensures optimal performance. It also includes regular updates to system features and database configurations to enhance functionality while maintaining compatibility with ever-changing technological standards. Data security measures, such as encryption and regular backups, can protect sensitive information from cyber threats. A well- maintained system will guarantee reliability and minimize downtime, offering users an interruption free and smooth experience.

User feedback plays a key role in refining the system. Integrating "Help" or "Contact Us" options allows users to share concerns, report issues, and suggest improvements. Promptly addressing user queries enhances trust and ensures a user-friendly experience. A dedicated support system can resolve technical difficulties efficiently, preventing disruptions in accessing the timetable. The

timetable management system can work perfectly, providing a reliable and effective solution for academic institutions, by prioritizing real-time updates, system maintenance, and user feedback. These features collectively contribute to an advanced scheduling system that is responsive, secure, and continuously improving to meet users' needs.

CHAPTER-5 OBJECTIVES

1. **Maximization of Allocation of Resources**

One must ensure conflict-free allocation of resources, i.e., there is no chance of a dispute regarding classrooms and teachers, machines, and materials. The tool must make it sure that none of the same events occur on the same area at the exact time unless compelling.

It equates resource allocation so that any form of misuse of the allocation can be limited, hence causing fewer idle time instances for each teacher and rooms.

2. **Conflict Resolution**

A key goal would be to minimize conflicts such as double-booking of classrooms or teachers, or scheduling students in two places simultaneously.

The algorithm should be capable of detecting potential conflicts and update the schedule while ensuring that it meets all conditions.

3. **Satisfaction of Conditions**

Availability of teachers: Only at specific hours or days could a teacher work.

Class room capacity: Classrooms do have a limited space and might have to accommodate a particular number of students.

Subject requirements: Some subjects require special classrooms and equipment.

Student groupings: Certain classes comprise specific groups of students, and their schedules have to be planned together.

It is important that all these constraints are respected while constructing the timetable.

4. **Flexibility and Adaptability**

Changes, which should be included in the generator, involve swapping classes and changes due to other unexpected causes, such as teacher illness or a room being unavailable.

Flexibility can allow timetables to be dynamic, helping the institution easily respond to unforeseen events.

5. **Fairness in distribution**

Ensure fairness in terms of equal number(s) of hours taught by a teacher, avoiding overloading or underloading.

For students, fairness can also mean balancing the number of classes per day to avoid fatigue (e.g., avoiding too many back-to-back classes or early morning sessions).

6. **Time Efficiency**

The timetable generation process should be quick, especially in larger institutions where there are numerous classes, teachers, and students involved.

An efficient algorithm should be able to produce a schedule in a reasonable time frame without excessive computational overhead.

7. **Minimization of Student and Teacher Stress**

The system should minimize the potential stress factors for students and teachers, like overloading a student's schedule or scheduling a teacher in back-to-back classes with little or no break time.

A balanced timetable can lead to better focus and overall well-being for both teachers and students.

8. **Automation and Scalability**

Automating the generation process removes the dependency on human intervention, which diminishes human error and labor.

The system should scale, meaning it could cope with different sizes of institutions, from small schools to large universities, without necessitating some kind of significant change in its architecture.

9. **User-Friendly Interface**

The system should be simple to use by administrators, easy interface for putting in data like teacher availability, student groups, etc., and reviewing the generated timetable.

CHAPTER-6 SYSTEM DESIGN & IMPLEMENTATION

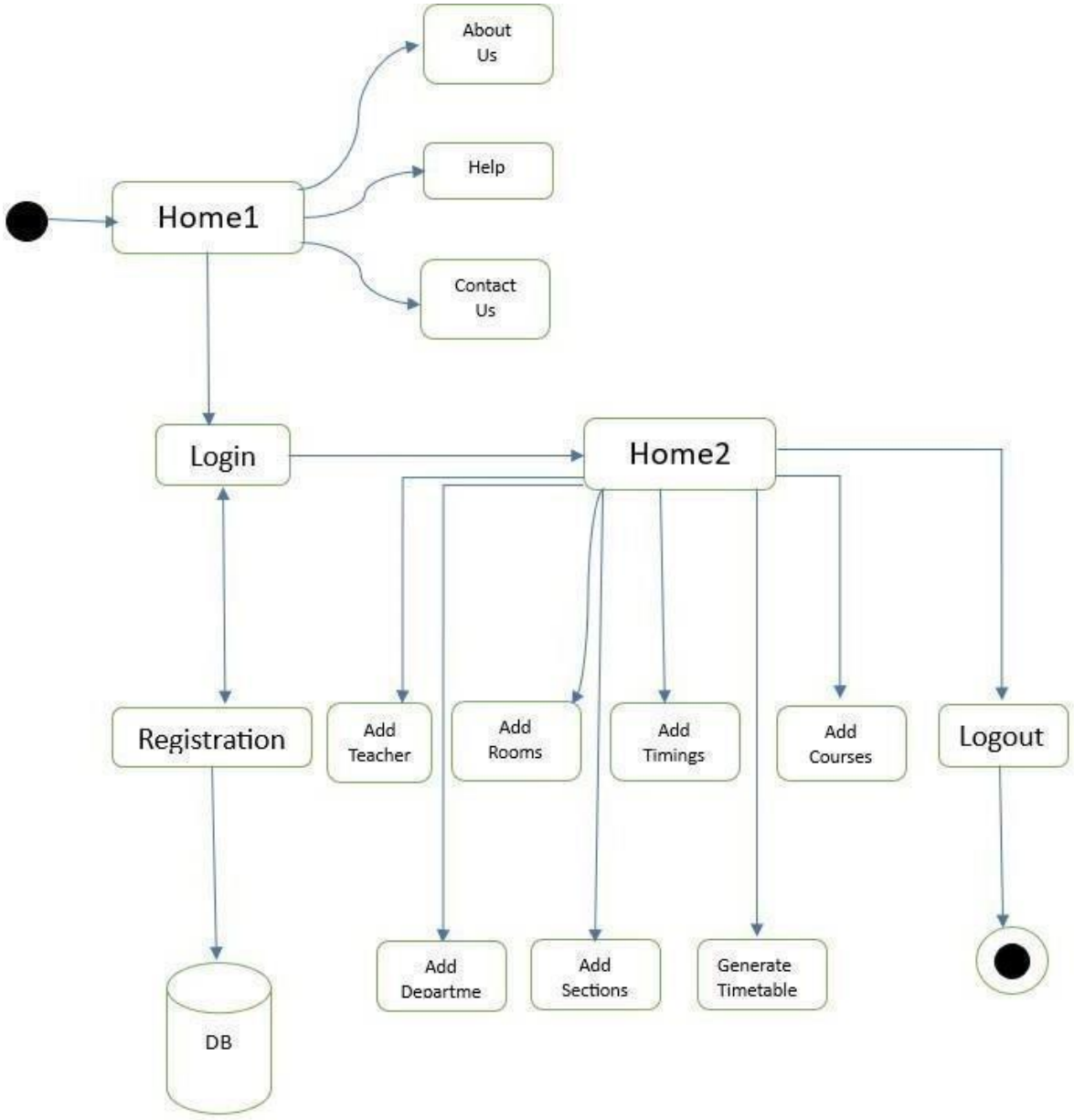


Fig 6.1: Architecture of Timetable Generation Application

The Proposed method consists of the following steps:

The system has been designed using three-tier architecture which includes the presentation layer (user interface) the application layer (business logic), and the data layer (centralized database). The system will

include the following modules: Login/Logout, Registration, Home, Add Teachers, Add Courses, Add Rooms, Add Timings, Generate Timetable, and Order Details.

Detailed Explanation of the Steps 6.1. Database Setup

The database is the core of this automatic timetable generation application, where all the vital information is either stored or fetched. SQLite3 is quite efficient to use while ensuring prompt and error-free data handling by direct query execution. SQLite3 is a lightweight yet robust tool for the management of institutional data. The database schema, therefore, accommodates:

Teacher Details: The name, department, qualifications, and availability are provided for each teacher.

Room Availability: Stores room numbers, seating capacities, and any special facilities (e.g., lab equipment).

Course Schedules: Maintains course codes, names, prerequisites, and credit hours.

Departmental Data: Links courses and teachers under specific departments.

Time Slots: Defines available time intervals for classes, ensuring no overlaps.

With proper indexing, normalization, and query optimization, the database supports efficient operations and rapid data retrieval, even with large datasets.

6.2. Home Pages (Home1 and Home2)

Home1 Page

This is the first landing page, intended for all users (admins, teachers, and students). Navigation to key sections, such as the following, is also intuitive.

About Us: Informs users about the application and its purpose in detail.

Help: They can get guidelines, FAQs, and tutorials to understand the system.

Contact Us: They can communicate with the support team for their queries and other technical issues.

Home2 Page:

The users will be redirected to the central dashboard, known as Home2, after successful login. It serves as a control hub for accessing key modules and functionalities like these mentioned:

Adding teachers, rooms, timings, and courses.

Managing departments and sections.

Generate timetables more efficiently.

Home2 is made dynamic and responsive, providing access across different devices while keeping the structure organized.

6.3. User Management

Login: The login functionality authenticates users, ensuring secure access to the system. Credentials are encrypted before being stored in the database, protecting sensitive information.

Registration: New users can register by providing essential details like name, email, and password. Registration data is securely stored in the database.

Logout: The logout feature terminates user sessions securely, clearing authentication tokens to prevent unauthorized access. This is especially crucial for protecting data on shared or public devices.

6.4. Administrative Functions

From the Home2 Dashboard, administrators have access to powerful tools to manage institutional data, including:

Add Teacher: Administrators can input teacher details, such as names, departments, and availability schedules.

Add Rooms: Enables administrators to register room details, including capacity and unique facilities like projectors or laboratory setups.

Add Timings: Time slots can be added or modified to ensure no scheduling conflicts. This includes defining periods, days, and durations.

Add Courses: Administrators can add courses with prerequisites, credit hours, and assigned instructors.

Add Departments: Organizes courses and teachers under relevant departments, making management and reporting easier.

Add Sections: Allocates specific student groups to courses, ensuring streamlined timetable generation.

6.5. Timetable Generation

The core functionality of the system lies in its ability to automatically generate optimized timetables. Using data from various modules (teachers, rooms, timings, and courses), the system ensures:

No Overlapping Time Slots: Validates teacher and room schedules to prevent conflicts.
Efficient Room Allocation: Assigns available rooms based on capacity and course requirements.
Balanced Workload Distribution: Distributes teaching responsibilities evenly among instructors, avoiding overburdening any individual. The generated timetables can be reviewed, edited, and exported for further use.

6.6. Security and Integration

Authentication
User authentication protocols (e.g., hashed passwords, session management) safeguard sensitive data during login and logout processes.
Django ORM
It interacts with the SQLite3 database so that any Django ORM can handle database queries about anything comfortably using Python.
Responsive Design
The web application is developed using responsive web design principles to ensure its accessibility and usability on desktops, tablets, and mobile devices.

CHAPTER-7 TIMELINE FOR EXECUTION OF PROJECT

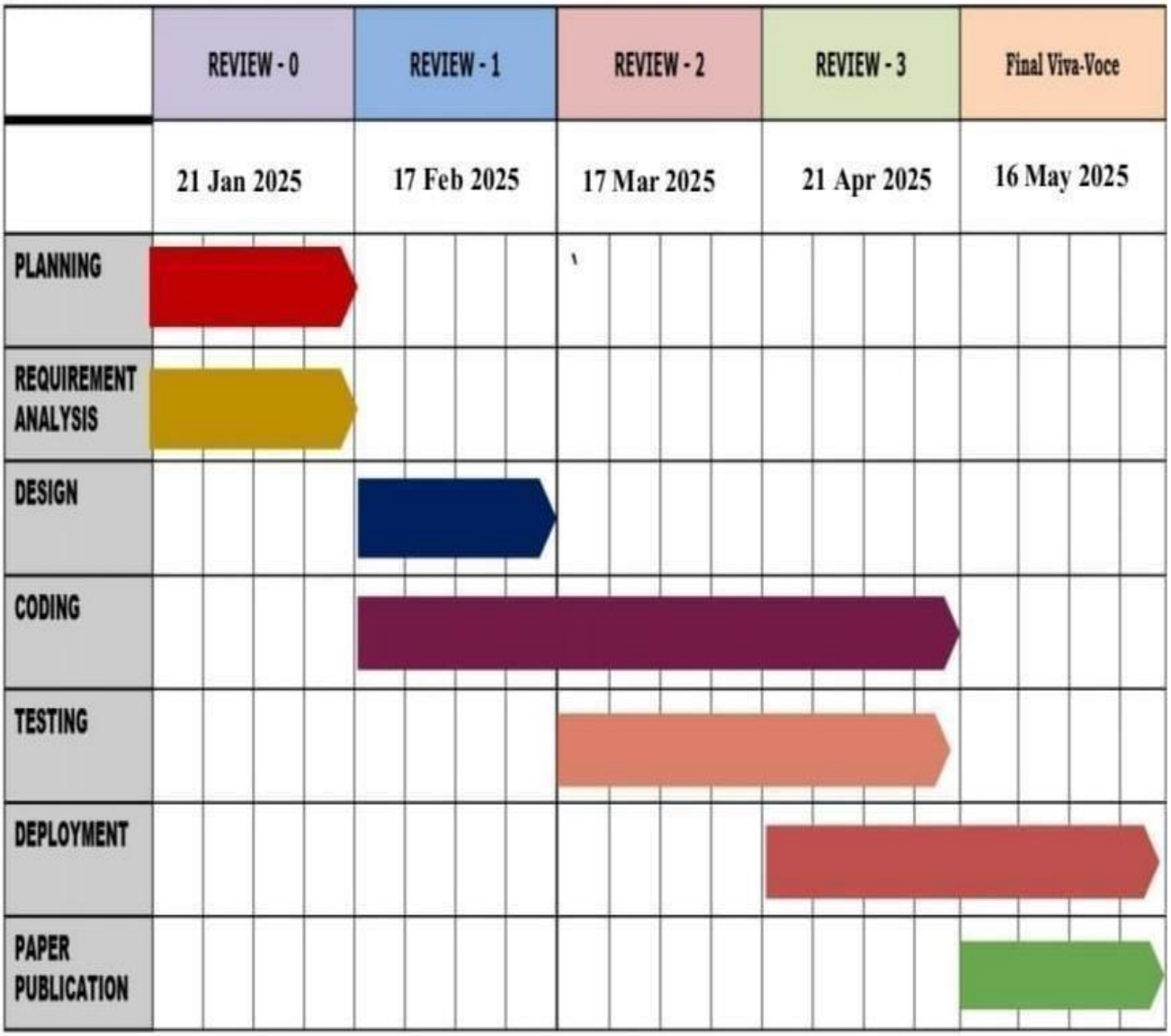


FIG.7.1 Timeline of The Project

CHAPTER-8 OUTCOMES

8.1. Time Efficiency

Manual creation and management of timetables take a lot of time, especially for large institutions with many courses, students, and faculty members. The automatic generation of timetables by the system reduces the time spent on these activities since scheduling and resource allocation are automated.

Important Features Contributing to Time Efficiency:

Automated Scheduling: The system draws data from the database and automatically creates a whole schedule in a matter of minutes, without manual input and alteration.

Bulk Data Management: Simultaneous input of multiple teachers, rooms, and courses will make setup much easier.

Real-time Modifications: Changes to schedules, such as adding a new course or modifying a time slot, are instantly reflected across the system.

Benefits:

Faster response times for last-minute changes or adjustments.

Saves ample time for the administrators to devote to other major tasks.

Prepares timetables ahead of time for the academic term, thus avoiding late bookings.

8.2. Errors

Errors are inherent when using manual scheduling. They include overlapping classes, breaking room capacities, and the simultaneous assignment of many tasks to a single instructor. The automated system eliminates this by allowing stringent validation checks during the generation of timetables.

Error-Prevention Mechanisms:

Conflict Detection: The system ensures that no two classes are assigned to the same room or instructor at the same time.

Capacity Management: Rooms are allocated based on their seating capacity, which prevents overcrowding.

Workload Limits: The system enforces pre-defined teaching workload limits so that faculty members are not overloaded.

Benefits:

Minimizes disruptions caused by schedule conflicts.

8.3. Increased Accessibility

A mobile application will facilitate real-time accessibility of schedules and notifications for faculty and administrators. This makes it accessible to anyone interacting with the communication and coordination process. Accessibility Features:

Application Friendly Design: It is responsive, and access can be gained on smartphones, tablets, and computers.

Push Notifications: Always alerting users regarding changes in schedule, cancellation, or other announcements.

Cloud Integration: Allows remote access to timetables from anywhere with an internet connection.

Benefits:

It ensures that the faculty and administrators are always aware of the updated schedules.

Reduces the number of printed timetables, thus promoting sustainability.

It provides flexibility for the users to access information on the go.

8.4. Balanced Workloads

It ensures that the teaching load is spread impartially among all faculty members, preventing burnout. It also keeps everyone at peak productivity based on availability, expertise, and limits of load.

Workload Management Features

Pre-set Limits: Administrators can predefine the maximum teaching hours for each faculty member.

Dynamic Adjustments: The system always adjusts to the eventual changes in faculty absence or courses added, maintaining a balance.

Graphical Reporting: Visualizing workloads aids administrators in identifying and managing overload.

Advantages:

Prevents overloading the faculty.

Resulting faculty morale and job satisfaction are higher.

Quality teaching will be delivered because no instructor will be overwhelmed with work.

Meets institutional guidelines on the allocation of work.

8.5. Scalability and Flexibility

It is developed with the idea that the system grows with the institution, encompassing more courses, faculty members, students, and rooms. The flexibility it exhibits ensures it adapts to evolving needs and changes in trends within education.

Scalability Capabilities:

Modular design: New functionality or modules can be added smoothly, such as online attendance recording or resource monitoring.

Optimal Performance: As the data input increases, this system does not lose its velocity.

Customized: The software can be changed to suit individual needs of institutes.

Adaptability Features

Policy Updates: The system integrates new scheduling policy or academic systems easily. Multiple Configurations: Supports different types of timetabling models: block scheduling as well as semester-based systems

Benefits

Keeps the usability of the system for a very long time.

Eliminates the requirement of frequent overhauls or replacement.

Facilitates growing demands of expanded institutions.

8.6. User Satisfaction

The effectiveness, reliability, and simplicity of the system contribute to an overall better experience of faculty, administrators, and students. Intuitive interface, automation, and no errors create higher user satisfaction.

User-Friendly Features:

Simple Navigation: Simple dashboards with easy-to-understand instructions will help the user navigate through the system easily.

Fast Deployment: There is little deployment time for a new user or institution.

Feedback Channels: The schedules can be requested to be modified or provided with feedback, promoting teamwork.

Benefits

Increased faculty and administrative acceptance

CHAPTER-9

RESULTS AND DISCUSSIONS

The Automatic Timetable Generation System provides a very efficient and user-friendly way of generating and managing institutional timetables. The system minimizes errors and optimizes the use of available resources by the automation of the scheduling process and the minimizing of effort in manual operations. It uses technologies such as HTML, CSS, JavaScript, Django, and SQLite3 to provide a highly strong and scalable solution to this problem. This chapter describes the results obtained by the system and its effects along with its limitations and possible future improvements.

Results

1. **Efficiency in Time Saving**

The program saves a tremendous amount of time for generating and managing timetables.
Automation: Assignments of time slots, conflict avoidance, and equal distribution of loads are done within seconds.
Effect: There is no manual interference, which conserves the most precious administrative hours.
Outcome: Institutions can strategize more on education rather than operational issues.
issues. This feature ensures timely access to healthcare without the need to travel long distances or navigate through busy healthcare systems, reducing delays in receiving medical attention.

2. **Conflict-Free Timetable Generation**

There will be no overlapping or overlapping of timetables generated in the system.
Validation Mechanisms: Internal checks prevent over-lapping time periods, over-booking rooms, or teachers' workload to be exceeded.
Impact: Error-free common with manual scheduling that is smooth and reliable.
Outcome: Increased confidence and reliability of the scheduling process, which works in favor of administrators, teachers, and students

3. **Ease of Accessibility**

The software provides real-time access to timetables to all stakeholders.

Responsive Design: The system is available on any gadget, from desktop to smartphones. Key Features: Teachers and administrators can at any given time view their timetables, edit them, or approve the updated ones.

Impacts: All stakeholders are always updated and up-to-date.

4. **Balanced Workloads**

The system distributes teaching work among faculty members in a balanced way.

Workload Distribution: The algorithms distribute classes in a manner that satisfies predefined workload limits.

Impact: Does not overload any one faculty member and makes sure that all the teachers are taught a reasonable number of classes.

Outcome: Faculty satisfaction goes up, and institutional policies are obeyed.

5. **Scalability and Adaptability**

The system is created with the dynamic necessities of institutions.

Scalability: The system will be able to support adding extra teachers, rooms, courses, or departments while maintaining performance.

Adaptability: The system can be designed to meet very specific institutional needs, such as different class-lengths or some departmental constraint.

Impact: Makes the application future-ready with long-term use.

6. **User Satisfaction**

The efficiency and reliability of the system and easy usability lead to higher user satisfaction.

Intuitive Interface: Very clean and simple design makes for easy navigation and operation.

Feedback Mechanism: The system can receive feedback from users for its development and evolution.

Outcome: This promotes the frequent use of the system and also helps to build confidence in the system.

Discussion

1. **Effect on Administrative Productivity**

The system saves a lot of administrative effort by automatically performing complicated scheduling operations.

Key Points: The process of assigning teachers, rooms, and resolving conflicts is automated and saves manual efforts.

Challenges: Proper data entry is crucial for producing accurate timetables.

2. **Error-Free Scheduling**

The system has strong validation mechanisms that ensure the generation of error-free timetables. Effectiveness: The system eliminates scheduling conflicts and ensures compliance with institutional policies.

Challenges: Unique constraints or exceptions may require more customization

3. **Real-Time Accessibility**

The application ensures that timetables are readily accessible to stakeholders.

Key Benefits: Teachers can view their schedules, administrators can make edits, and students can check their classes in real time.

Challenges: Ensuring consistent uptime and performance during peak usage periods

4. **Integration with Institutional Systems**

The system can be integrated with other institutional platforms for smooth operation.

Opportunities: Integration with attendance tracking or academic management systems can enhance functionality.

Challenges: Compatibility with legacy systems and data synchronization are potential hurdles.

5. **Data Privacy and Security**

Protecting institutional and personal data is a critical aspect of the application.

Privacy Measures: Secure login, encrypted database storage, and compliance with data protection policies ensure data security.

Challenges: Preventing unauthorized access and safeguard against data breaches need constant monitoring and updates.

6. **Scalability and Future Enhancements**

The system is designed to accommodate large institutions and be responsive to the changing needs. Future Features: Integration of features like AI-based timetable optimization, automated notification systems, or multi-campus support.

Challenges: Balancing feature expansion with system performance and usability.

7. **Cost-Effectiveness and Affordability**

The system helps institutions save on costs by minimizing manual effort and optimizing resources. For Institutions: Cost savings in administrative time and effort get reflected in lower operations. For Users: Time saved in accessing and managing their schedules contributes to efficiency as a whole Challenges: Making it affordable for smaller institutions which have limited budgets.

8. **User Feedback and Continuous Improvement**

User input and usage analytics will guide the development of the system.

Feedback Mechanism: The system gets updated regularly based on user input so it remains relevant and effective.

Continuous updates: adding features and improving performance based on user needs and technological advancements.

Challenges: Managing user expectations while maintaining a streamlined development process.

CHAPTER-10 CONCLUSION

The proposed The Automatic Timetable Generator brings an unmatched level of flexibility and precision to the scheduling process. The software ensures that every timetable is not only conflict-free but also optimally tailored to the needs of all stakeholders by integrating factors like faculty availability, room constraints, and institutional policies into its algorithm. The application allows for a teaching environment that avoids overburdening faculty members with too many classes, ensures each classroom is utilized to the fullest extent, and holds administrative staff to perform strategic tasks meant to maintain an academically effective and harmonious environment.

Scalability allows it to be used for any educational institution of size, from a small school to a giant university. Whether one is dealing with handfuls of faculty members or thousands, this Automatic Timetable Generator will be able to handle all the demands an academic institution might have on it. This ensures that no matter what changes occur in educational institutions, this system will be relevant and useful.

Looking ahead, predictive analytics will further enhance the capabilities of the platform by allowing institutions to anticipate and proactively resolve scheduling issues before they even arise. This forward-thinking approach ensures that the system not only responds to current needs but also adapts to future trends and challenges in academic scheduling.

In the final analysis, the Automatic Timetable Generator is far more than just an automatic timetable generator-it is a holistic solution that transforms the overall academic planning approach of an institution. By automatically using real-time data and high-end optimization techniques, it equips educational institutions with a better ability to run their affairs and facilitates faculty and administrative staff to concentrate on what matters: delivering quality education. This is a move that represents one giant leap into the future for academic scheduling; powerful, reliable, and user-centric, which makes it work well for anyone who is engaged in the education process.

REFERENCES

[1]. Tavakkol, M., & Parsa, M. (2021). A Hybrid Genetic Algorithm for University Course Timetabling Problem Considering Faculty Preferences. *Computers & Industrial Engineering, 157*, 107327.
[https://doi.org/10.1016/j.cie.2021.107327]

[2]. Rong, Q., & Lee, K. (2022). Multi-Objective Optimization for University Timetabling Problem: A Comparative Study of Algorithms. *Journal of Scheduling, 25*(1), 57-72. [https://doi.org/10.1007/s10951-021-00788-3]

[3]. Hassan, M., & Khalil, M. (2023). An Intelligent Course Scheduling System Using Machine Learning Techniques. *Journal of Educational Computing Research*, *61*(3), 445-465. [https://doi.org/10.1177/07356331221122514]

[4]. 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]

[5]. 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]

[6]. 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]

[7]. Akı, O. (2020). University Course Timetabling Using Genetic Algorithms. In Proceedings of the International Scientific Conference UNITECH'2020, Gabrovo, Bulgaria. [https://doi.org/10.13140/RG.2.2.25215.43686]

[8]. 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://doi.org/10.48550/arXiv.2212.02230]

APPENDIX-A PSEUDOCODE

```
Start
    // Navigation to Home1
Display options: About Us, Help, Contact Us.
Login
If User selects "About Us":
Display system details and its purpose
If User selects "Help": Display
help page or FAQs If User selects
"Contact Us":
Display contact details for technical support If
User selects "Login": Navigate to
Login Page

// Login Process
Accept user name and
password

Verify credentials with
database If
valid:
    Redirect to
Home2 Else:
    Display "Invalid credentials"

// Registration Process
    Accept registration details (name, email, password, etc.)
    Insert registration details into the database
Display "Registration successful"

// Navigate to Home2 (Post Login)
```

Display options: Add Teacher, Add Rooms, Add Timings, Add Courses, Add Department, Add Sections, Generate

Timetable

If User selects "Add Teacher":

Accept teacher details (ID, name, department, etc.)
Validate details
Insert teacher details into database

If User selects "Add Rooms":

Accept room details (room ID, name, capacity, etc.)
Validate details
Insert room details into

database

If User selects "Add Timings":

Accept timing details (start time, end time, session type, etc.)
Validate details
Insert timing details into

database

If User selects "Add Courses":

Accept course details (course ID, name, semester, etc.)
Validate details
Insert course details into database

If User selects "Add Department":

Accept department details (department ID, name, head of department, etc.)
Validate details
Insert department details into database

If User selects "Add Sections":

Accept section details (section ID, course ID, etc.)
Validate details
Insert section details into database

If User selects "Generate Timetable":

Fetch teacher, room, course, and timing details from the database
Generate timetable based on availability and constraints

```
// Logout Process
   If User selects "Logout":
      End session and return to Home1
End
```

APPENDIX-B SCREENSHOTS

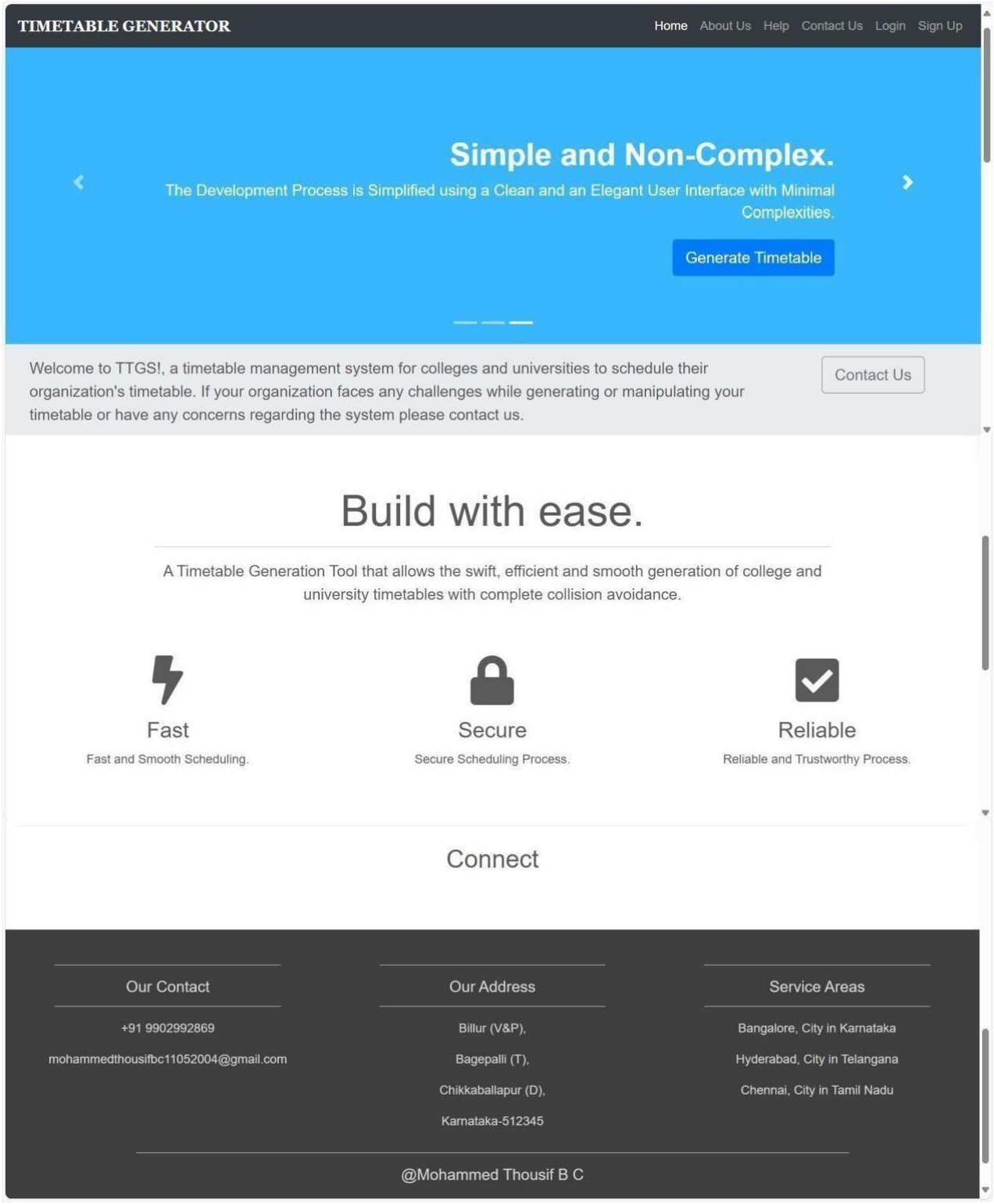


Fig.1.Home1 Page

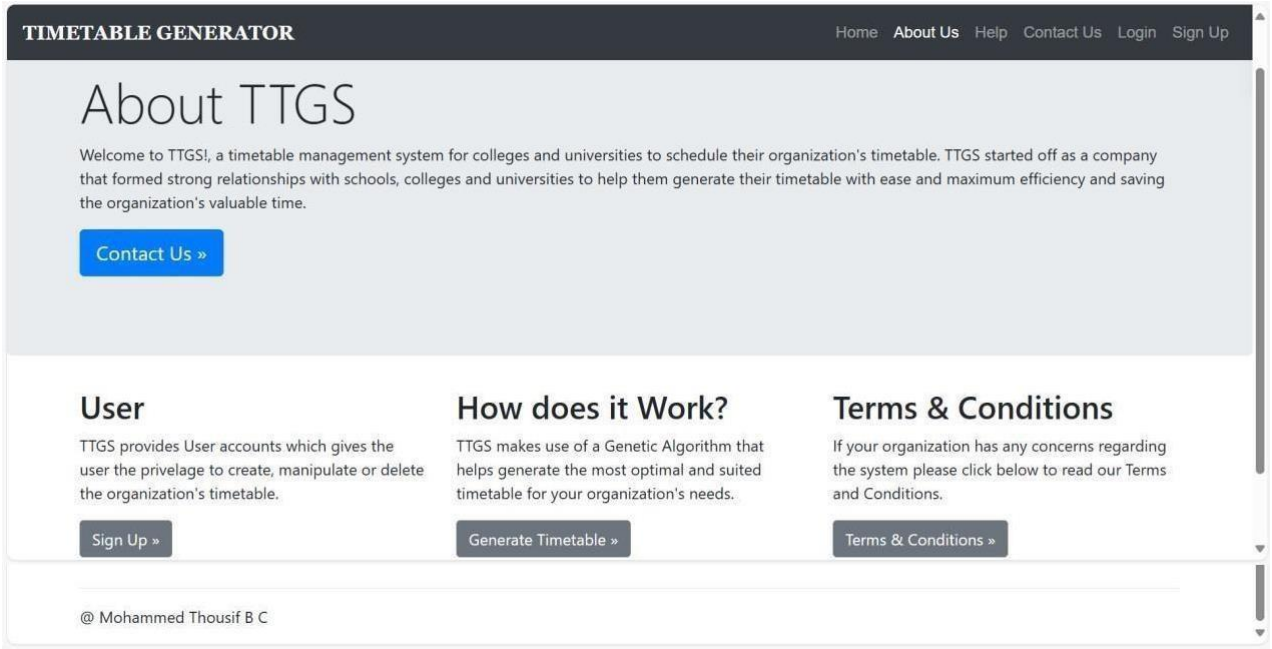


Fig.2.About Us Page

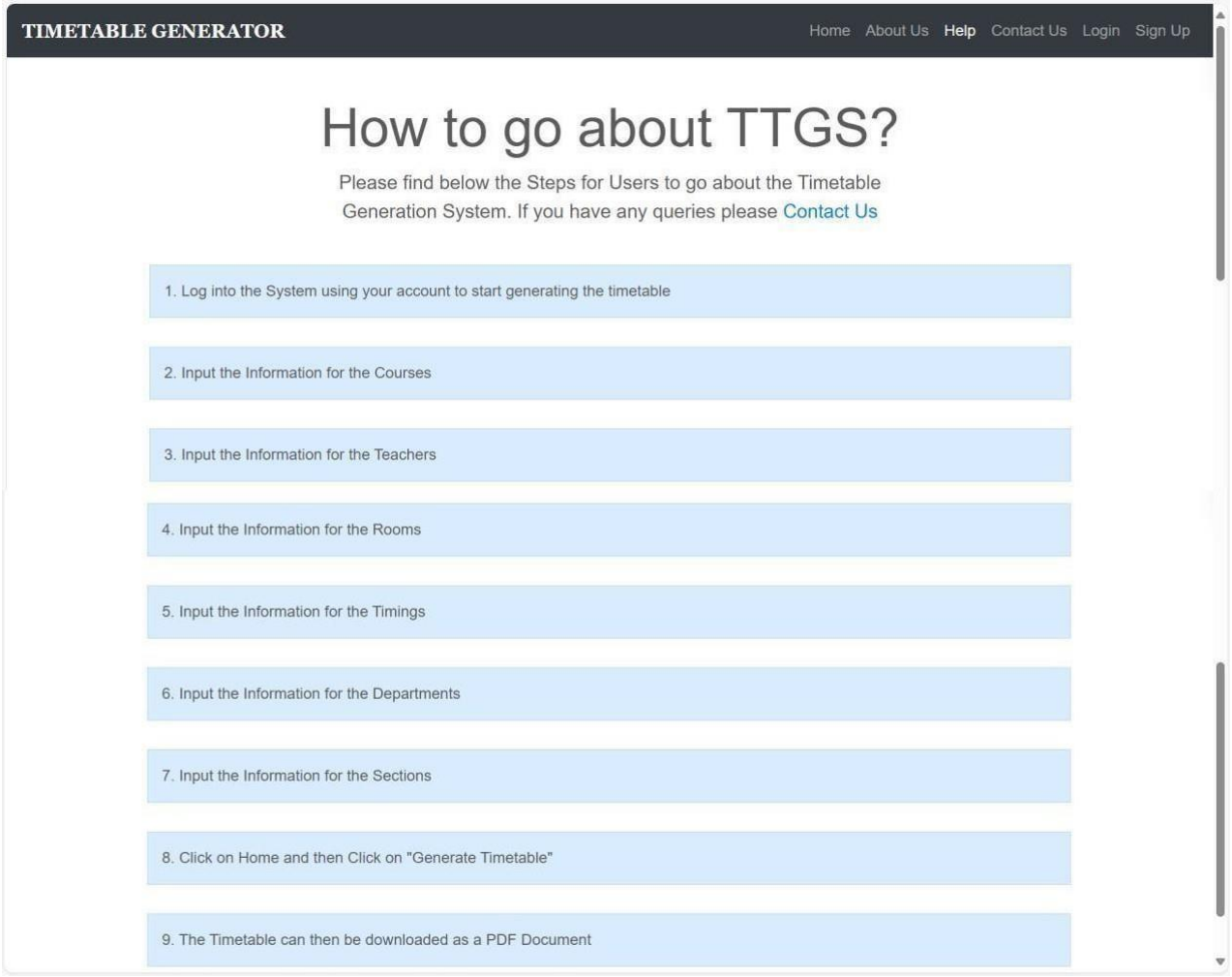


Fig.3.Help Page

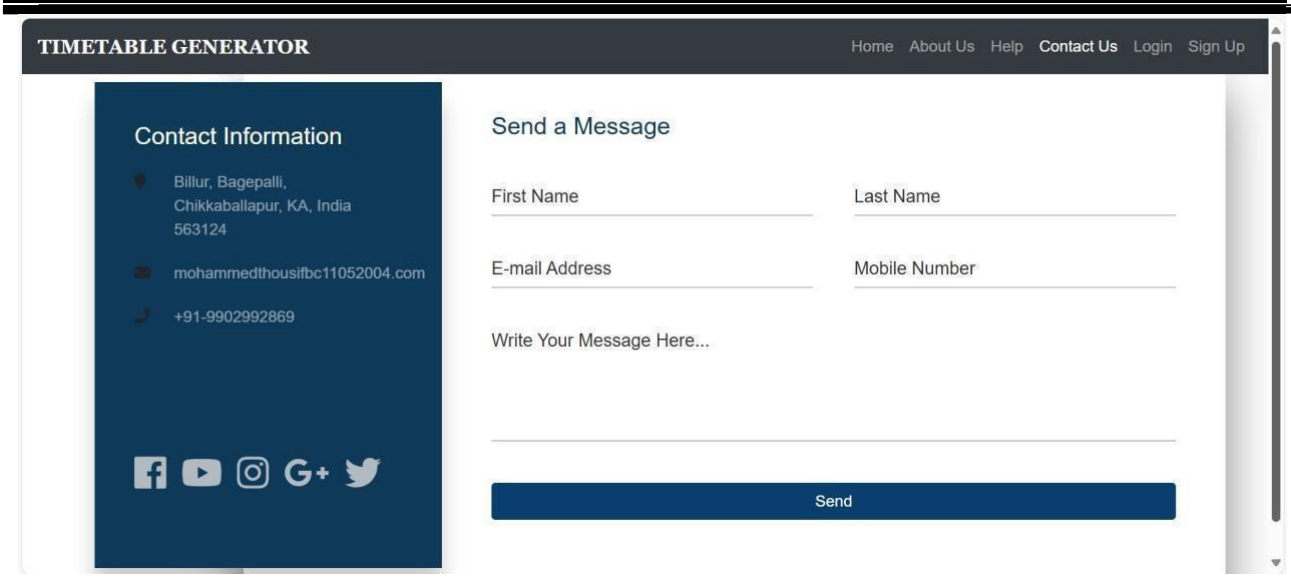


Fig.4.Contact Us Page

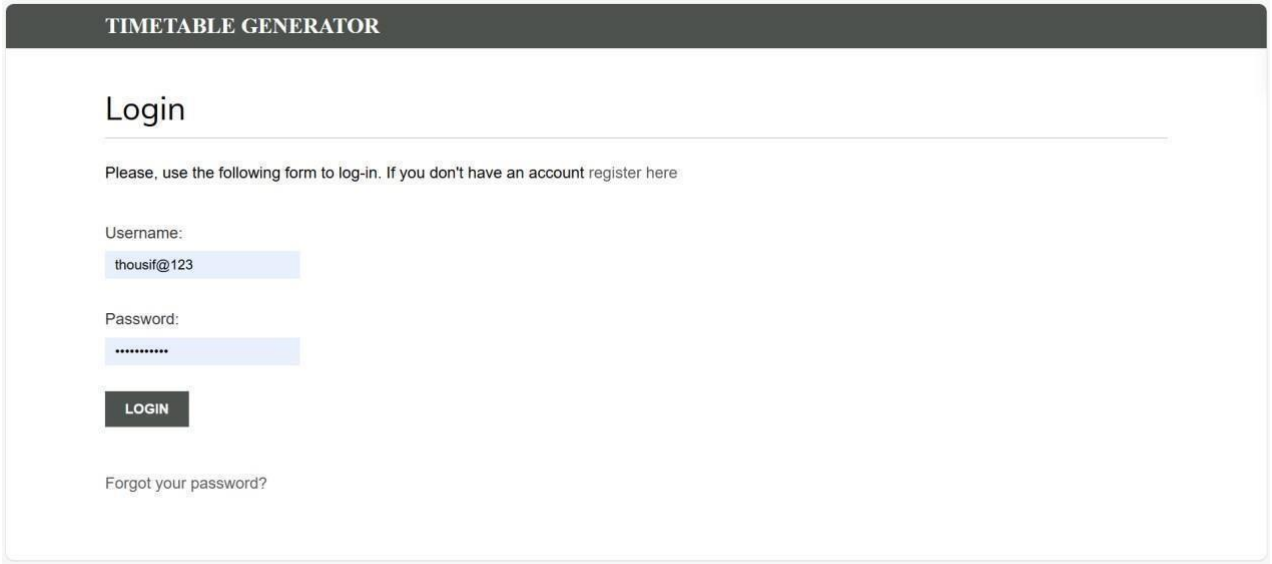


Fig.5.login Page

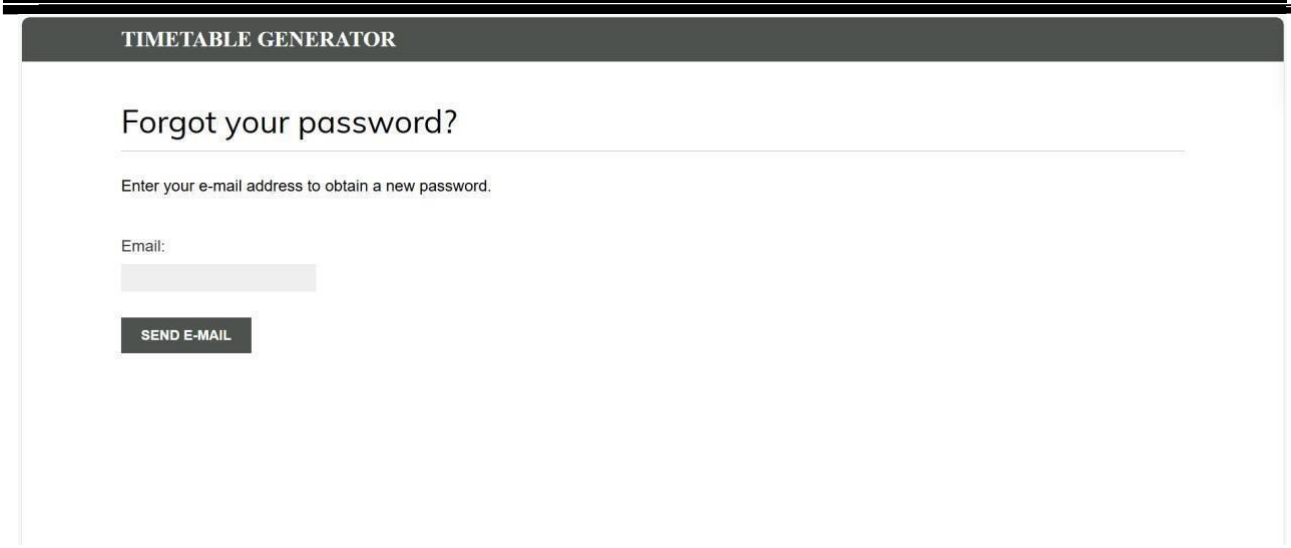


Fig.6.Forgot password Page

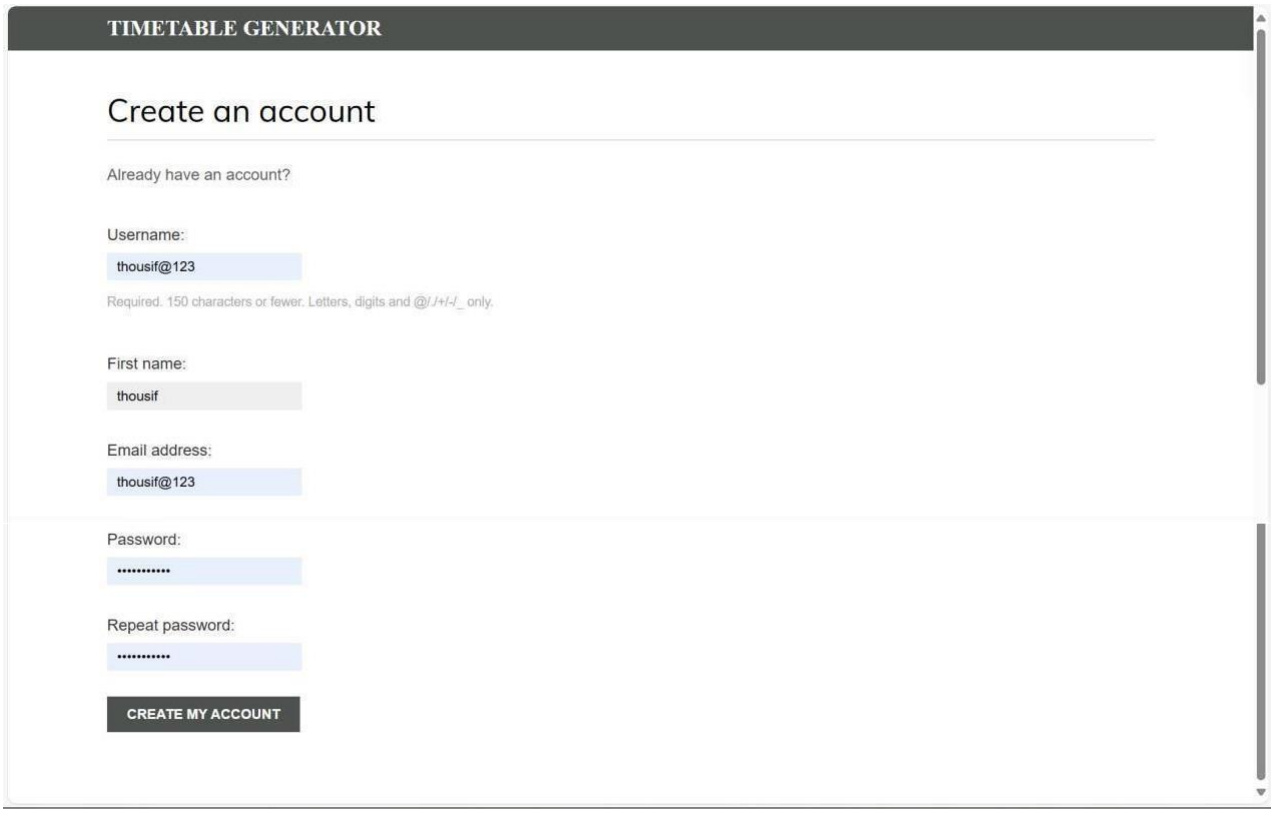


Fig.7.SignUp Page

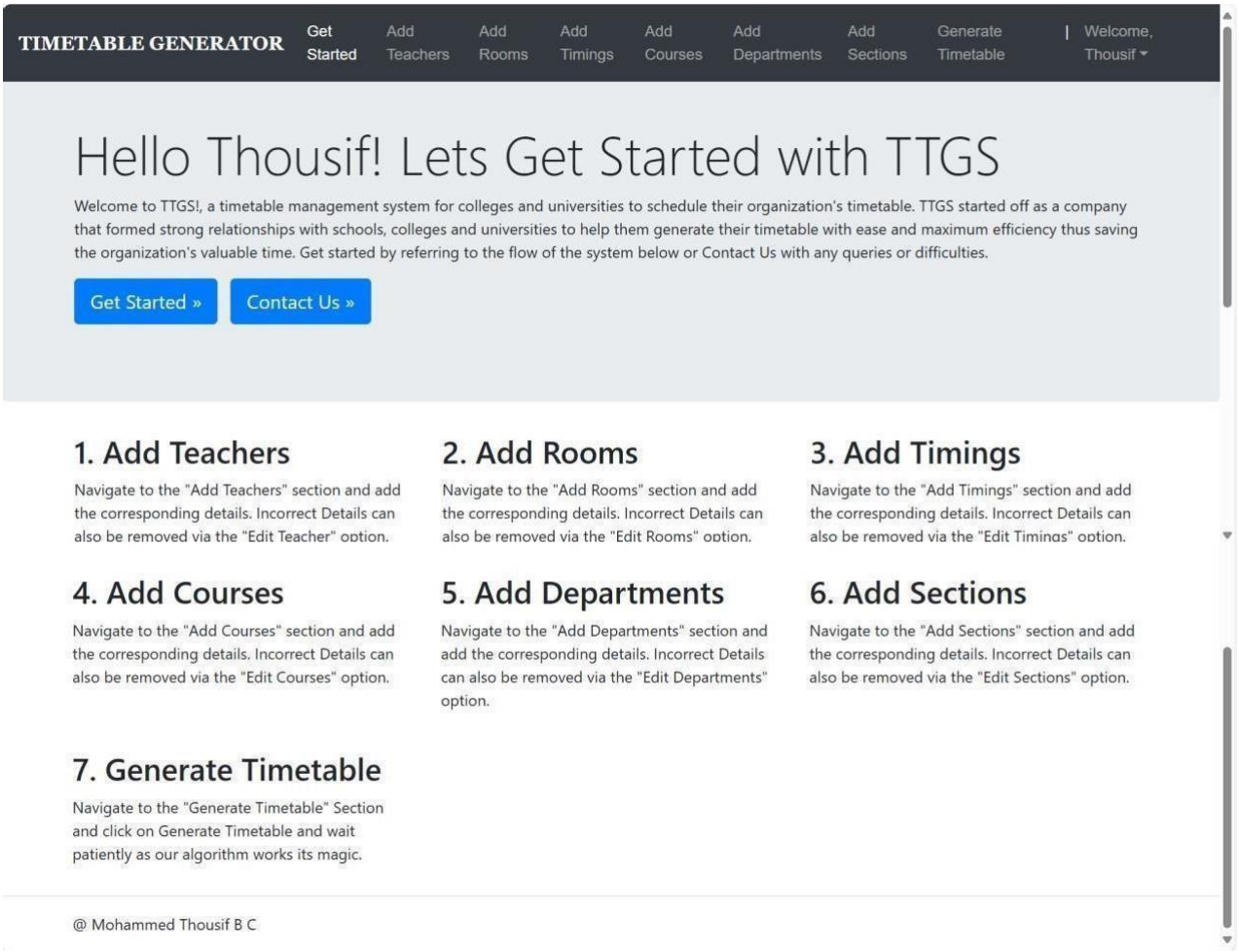


Fig.8.Home2 Page

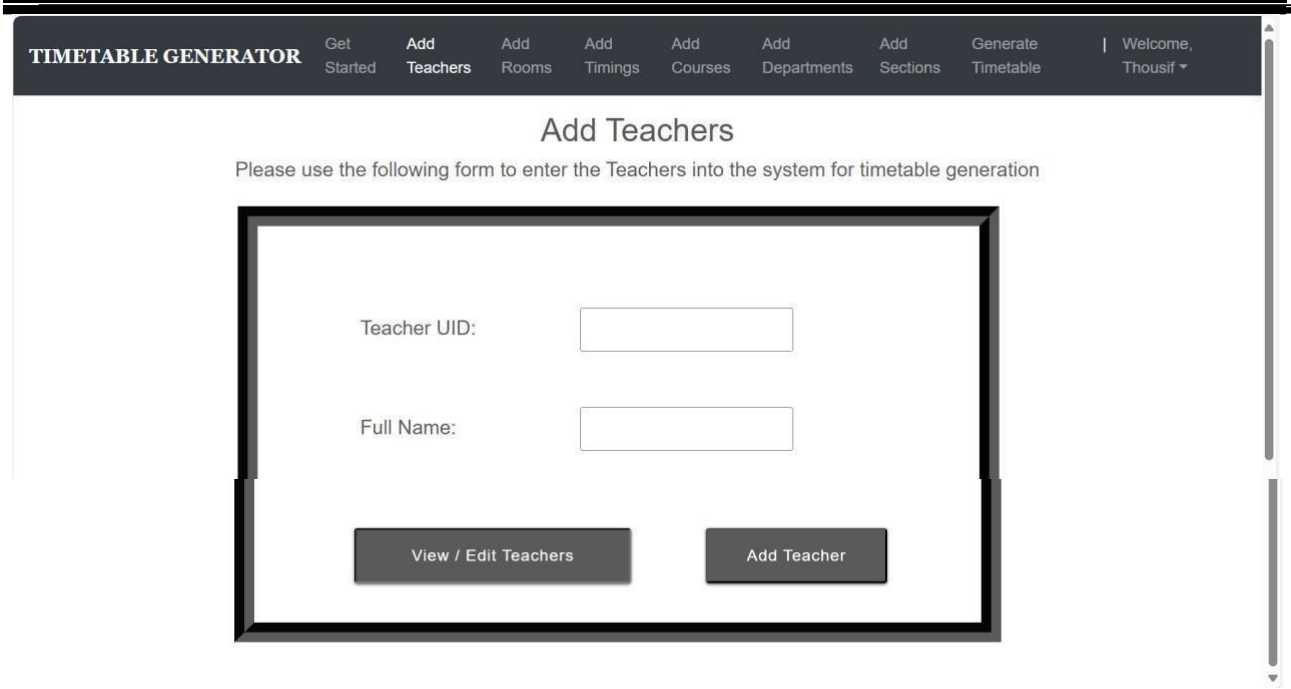


Fig.9.Add Teachers Page

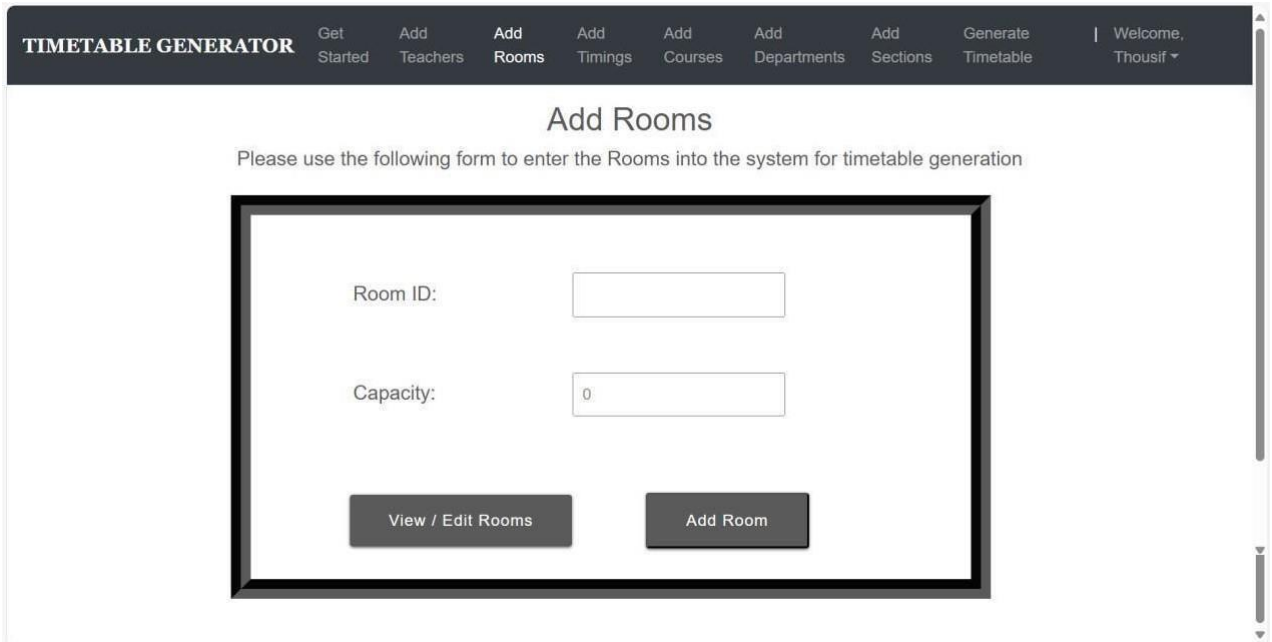


Fig.10.Add Rooms Page

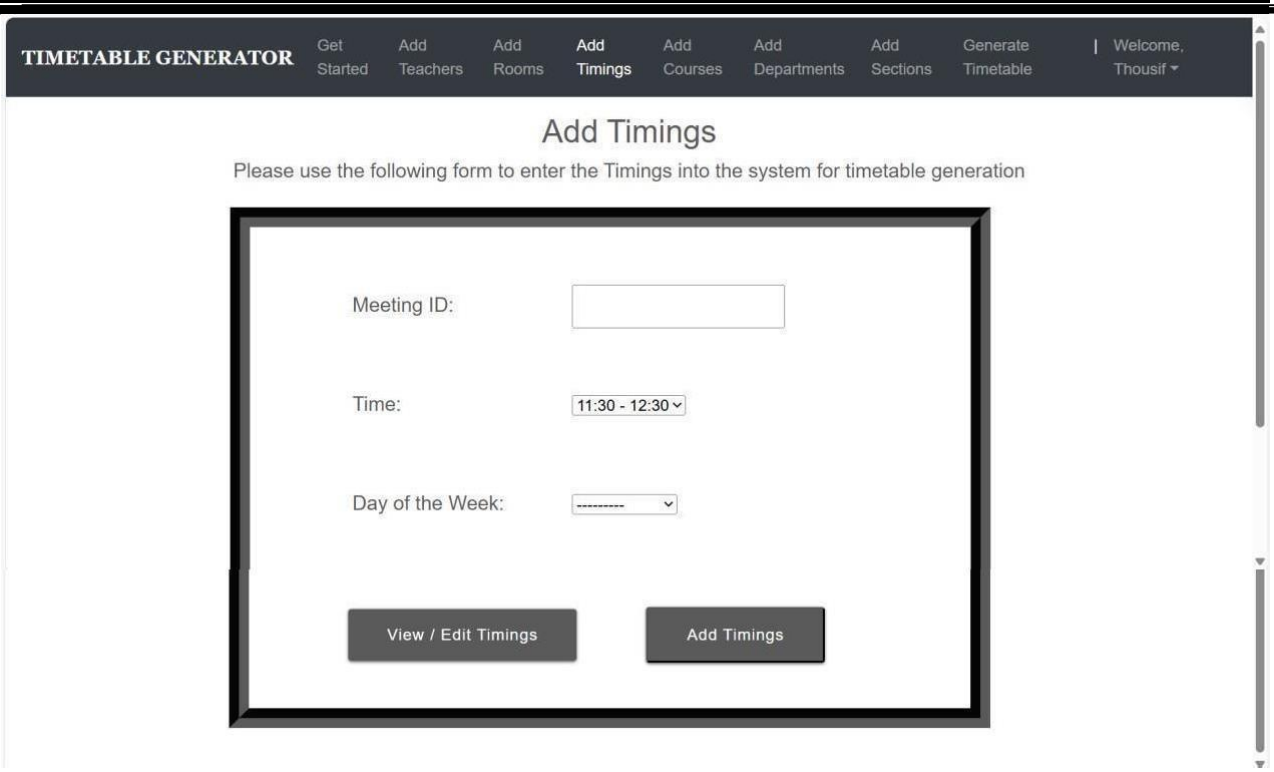


Fig.11.Add Timings Page

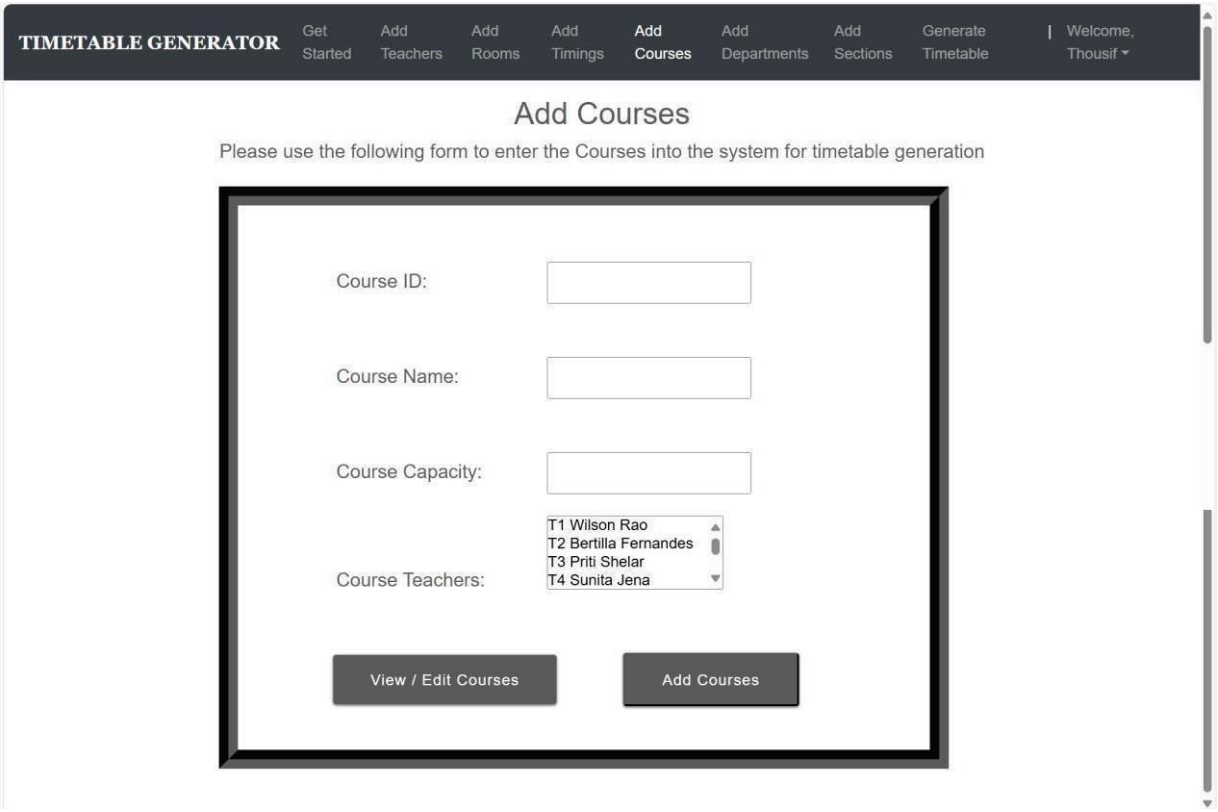


Fig.12.Add Courses Page

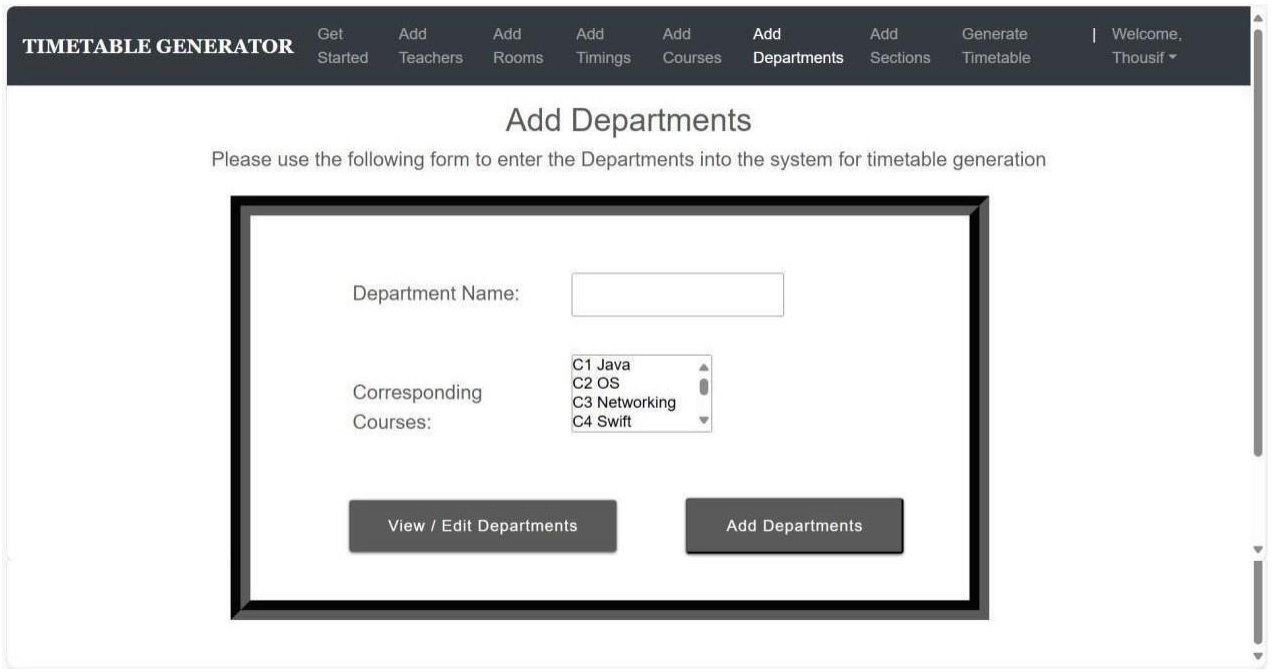


Fig.13.Add Departments Page

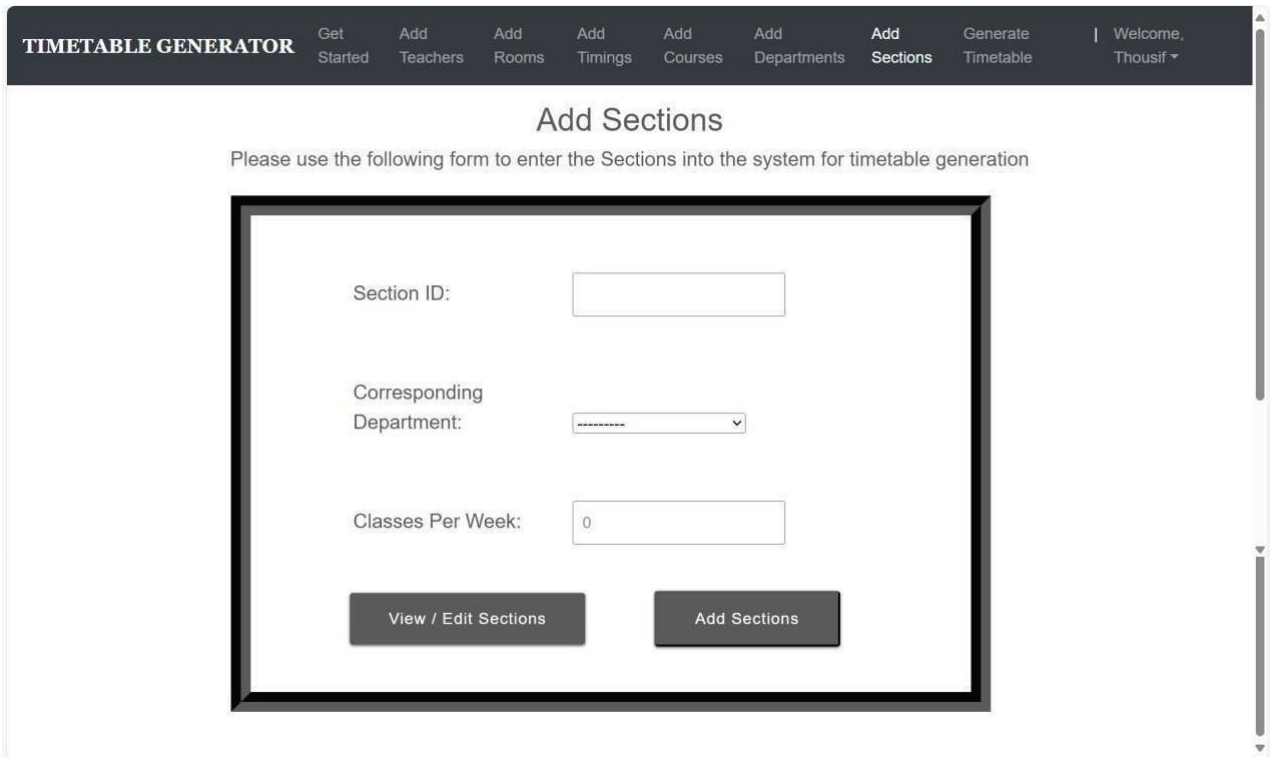


Fig.14.Add Sections Page

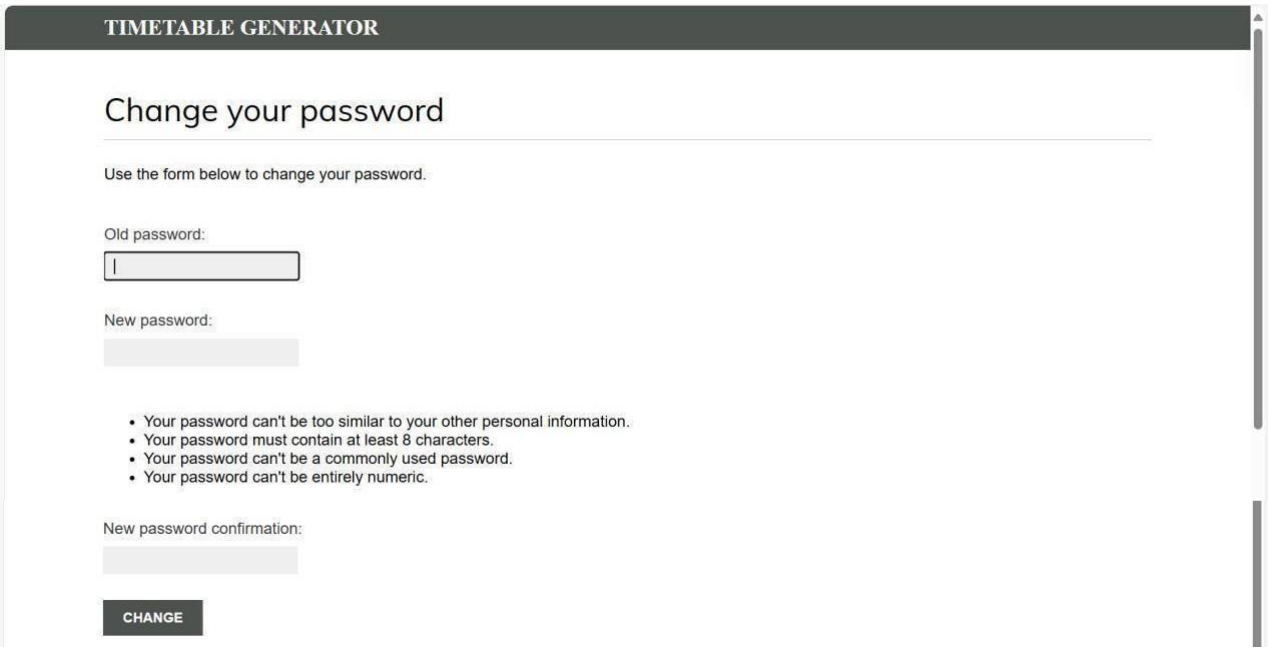


Fig.15.Change Password Page

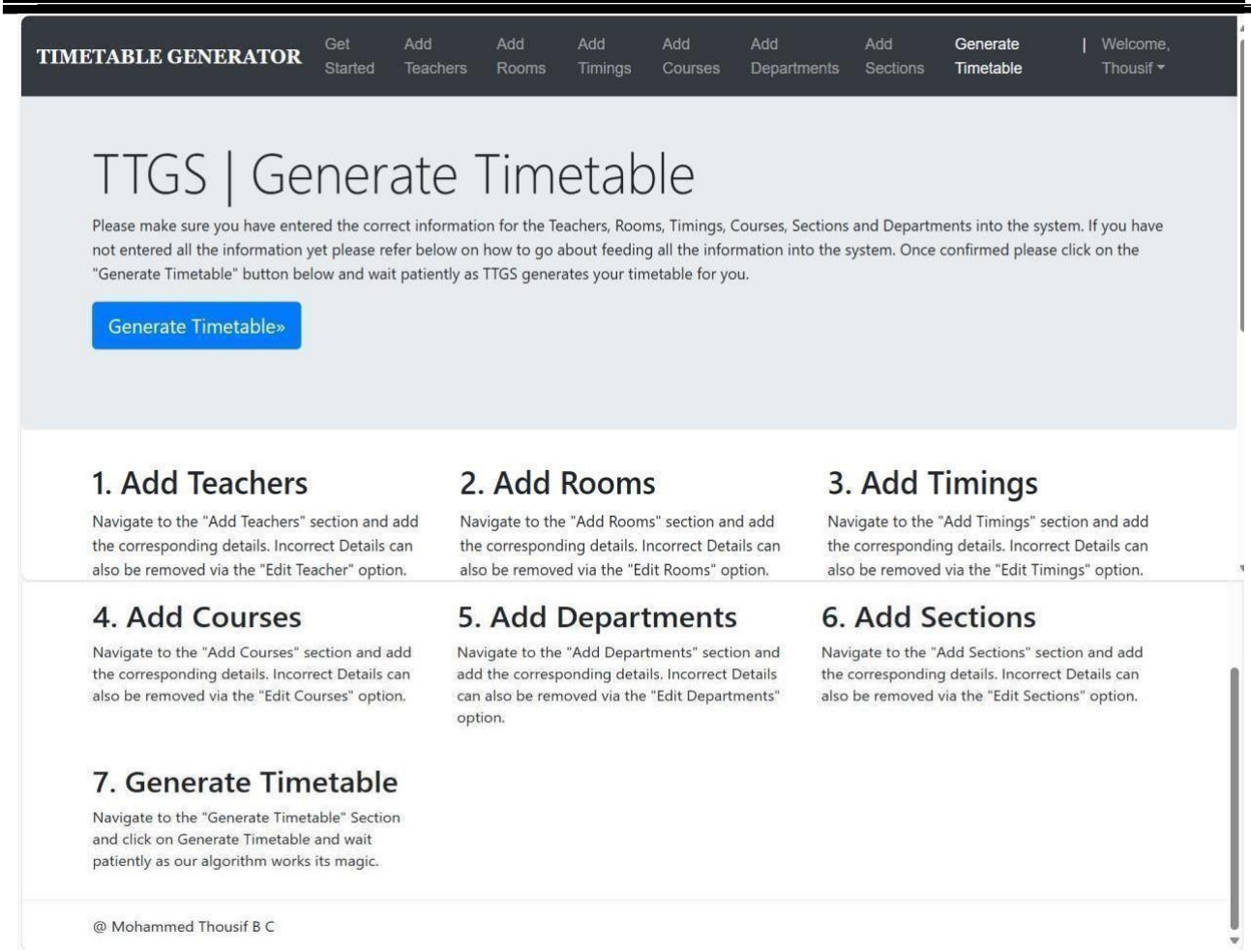


Fig.16.Generate Timetable Page

Fig.17.output Page

APPENDIX-C ENCLOSURES

Fig.18.Summer-term timetable generation

Our summer-term timetable generation project aligns very strongly with two United Nations Sustainable Development Goals — Quality Education (SDG 4) and Industry, Innovation, and Infrastructure (SDG 9).

Our primary goal is to streamline and enhance the process of constructing class schedules, particularly during the brief but intense summer term. Through the utilization of technology to construct efficiently organized timetables automatically, we ensure students have equitable access to classes without conflicts or misunderstandings. It also simplifies the planning process for teachers to manage their time and concentrate on teaching.

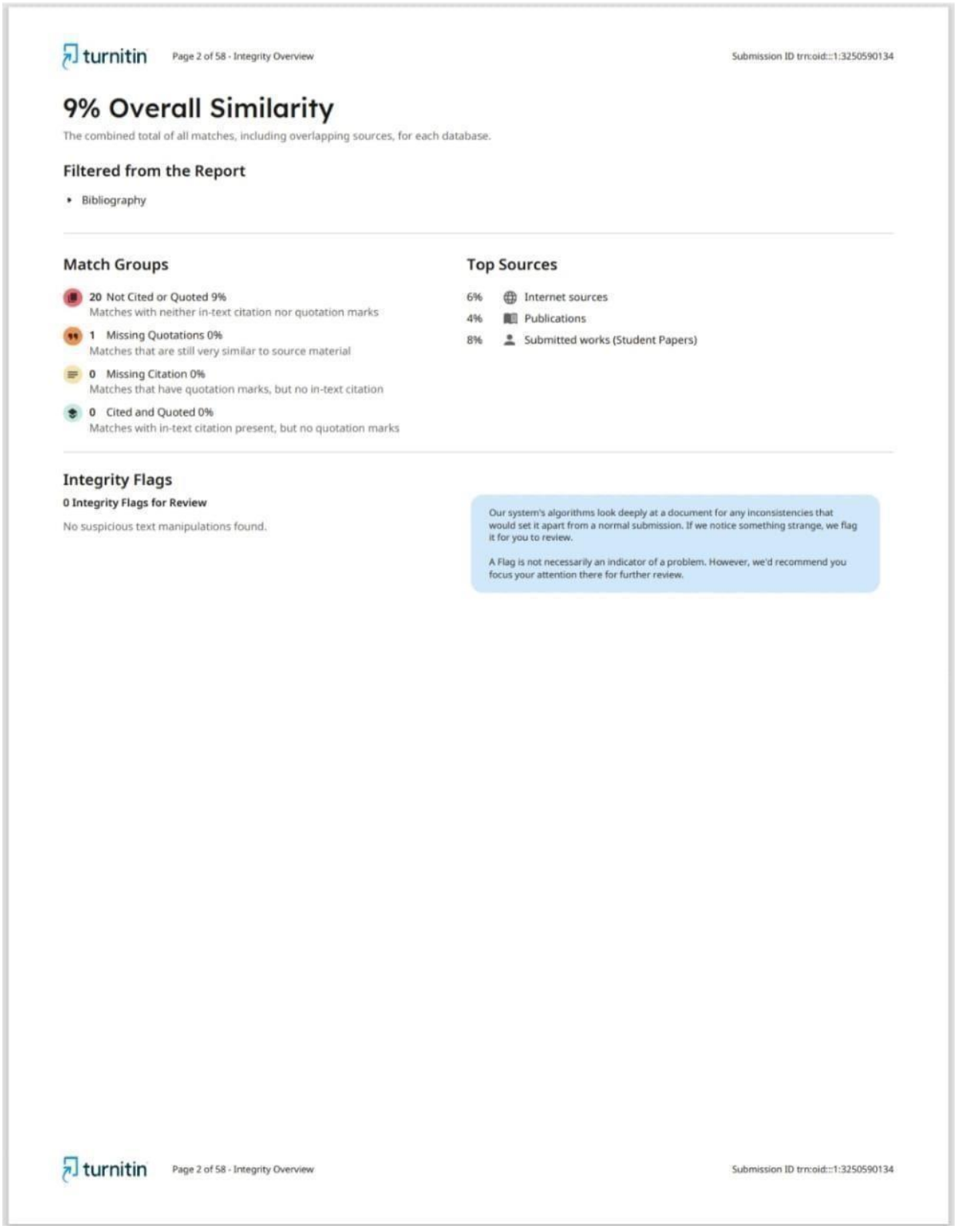
In addition, the project innovates by using contemporary tools to address a common issue in education. This not only saves time but also enables institutions to function more efficiently and smoothly. In this process, our work assists in the creation of smarter educational infrastructure and aids in ensuring equal learning opportunities for all.

APPENDIX-C
ENCLOSURES


DOI: 10.55041/IJSREM47904
IJSREM
e-Journal
ISSN: 2582-3930
Impact Factor: 8.586
INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT
An Open Access Scholarly Journal || Index in major Databases & Metadata
CERTIFICATE OF PUBLICATION
International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to
Ravi Shivaji Mahipati
in recognition to the publication of paper titled
Efficient and Automated Approach for Time Table Generation
published in IJSREM Journal on Volume 09 Issue 05 May, 2025
www.ijsrem.com
Editor-in-Chief
IJSREM Journal
e-mail: editor@ijsrem.com


DOI: 10.55041/IJSREM47904
IJSREM
e-Journal
ISSN: 2582-3930
Impact Factor: 8.586
INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING & MANAGEMENT
An Open Access Scholarly Journal || Index in major Databases & Metadata
CERTIFICATE OF PUBLICATION
International Journal of Scientific Research in Engineering & Management is hereby awarding this certificate to
V S Krishna Chaitanya Avvari
in recognition to the publication of paper titled
Efficient and Automated Approach for Time Table Generation
published in IJSREM Journal on Volume 09 Issue 05 May, 2025
www.ijsrem.com
Editor-in-Chief
IJSREM Journal
e-mail: editor@ijsrem.com

REPORT PLAIGARISM



turnitin Page 2 of 58 - Integrity Overview Submission ID trn:oid:::1:3250590134

9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Match Groups

- 20 Not Cited or Quoted 9%
  Matches with neither in-text citation nor quotation marks
- 1 Missing Quotations 0%
  Matches that are still very similar to source material
- 0 Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- 0 Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

6% Internet sources
4% Publications
8% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

turnitin Page 2 of 58 - Integrity Overview Submission ID trn:oid:::1:3250590134

Shet Reshma Prakash

Shet_Reshma_Prakash_tarun_Research_paper_0385_1

Quick Submit

Quick Submit

Presidency University

Document Details

Submission ID

trn:oid:::1:3248592022

Submission Date

May 13, 2025, 11:53 AM GMT+5:30

Download Date

May 13, 2025, 12:01 PM GMT+5:30

File Name

Shet_Reshma_Prakash_tarun_Research_paper_0385_1.docx

File Size

54.6 KB

4 Pages

2,540 Words

14,980 Characters

2% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography

Match Groups

4 Not Cited or Quoted 2%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

2% Internet sources
1% Publications
1% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **4** Not Cited or Quoted 2%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

2% Internet sources
1% Publications
1% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Student papers	
University of Wales Swansea		1%
2	Internet	
ijsrem.com		<1%
3	Internet	
research-repository.griffith.edu.au		<1%
4	Internet	
courses.engr.illinois.edu		<1%

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

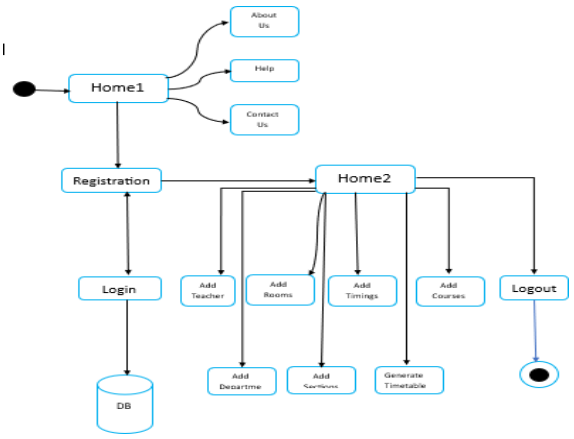


Figure 1. Architecture of Timetable Generation Application

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].