AN APPLICATION TO GENERATE SUMMER-TERM TIMETABLE

A PROJECT REPORT

Submitted by,

Mr. Mohammed Thousif B C - 20211CSE0388

Mr. Ullas H R - 20211CSE0299

Under the guidance of,

Mr. Md Ziaur Rahman

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

PRESIDENCY UNIVERSITY

BENGALURU

MAY 2025

PRESIDENCY UNIVERSITY

PRESIDENCY SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project report **"AN APPLICATION TO GENERATE SUMMER-TERM TIMETABLE"** being submitted by Mohammed Thousif B C, Ullas H R, bearing roll numbers "20211CSE0388", "20211CSE0299" in partial fulfillment 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.

Mr. MD ZIAUR RAHMAN
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-Vice Chancellor - 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 **An application to Generate Summer-Term Timetable** in partial fulfillment 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 **Mr. Md Ziaur Rahman, 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
Mohammed Thousif B C	20211CSE0388	
Ullas H R	20211CSE0299	

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. Md Ziaur Rahman, Assistant Professor** and Reviewer **Ms. Akkamahadevi, 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.

Mohammed Thousif B C

Ullas H R

LIST OF TABLES

LIST OF FIGURES

CONTENTS

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 labor-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 confict 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 analyze 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

Tavakkol, M., & Parsa, M. *A Hybrid Genetic Algorithm for University Course Timetabling Problem Considering Faculty Preferences.*

Computers & Industrial Engineering, 157, 107327. Proposes a hybrid genetic algorithm to solve university course timetabling problems by considering faculty preferences.

Confronts the challenges of optimizing timetabling in relation to faculty availability and preferences, while still meeting the constraints.

Rong, Q., & Lee, K. *Multi-Objective Optimization for University Timetabling Problem: A Comparative Study of Algorithms.*

Journal of Scheduling, 25(1), 57-72. Performs a comparative study of multi-objective optimization algorithms for university timetabling problems.

Compares different algorithmic approaches in light of multiple objectives for timetabling, which can be conflicting, such as minimizing resource usage and reducing conflicts.

Wang, X., & Xu, H. *A Novel Memetic Algorithm for Solving University Timetabling Problems.* **Expert Systems with Applications**, Volume 178, Article 115018. Proposes a new memetic algorithm to solve the university timetabling problem in an efficient way.

Faces challenges in optimizing the algorithm for large datasets and complex timetabling constraints with computational efficiency.

Hassan, M., & Khalil, M. *An Intelligent Course Scheduling System Using Machine Learning Techniques.*

Journal of Educational Computing Research, 61(3), 445-465. Proposes an intelligent course scheduling system making use of machine learning to optimize the course scheduling process.

Faces challenges in properly integrating machine learning algorithms with scheduling systems in a way that achieves accurate predictions about resource requirements.

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 canceled 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 favorite 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 enrollment 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 enrollment, 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 enrollment in courses and real-time changes in the program based on the course enrollment 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.

- **Data Privacy and Security**

Given that automatic timetable systems deal with sensitive information, including student enrollment 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 enrollments 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 enrollment 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:

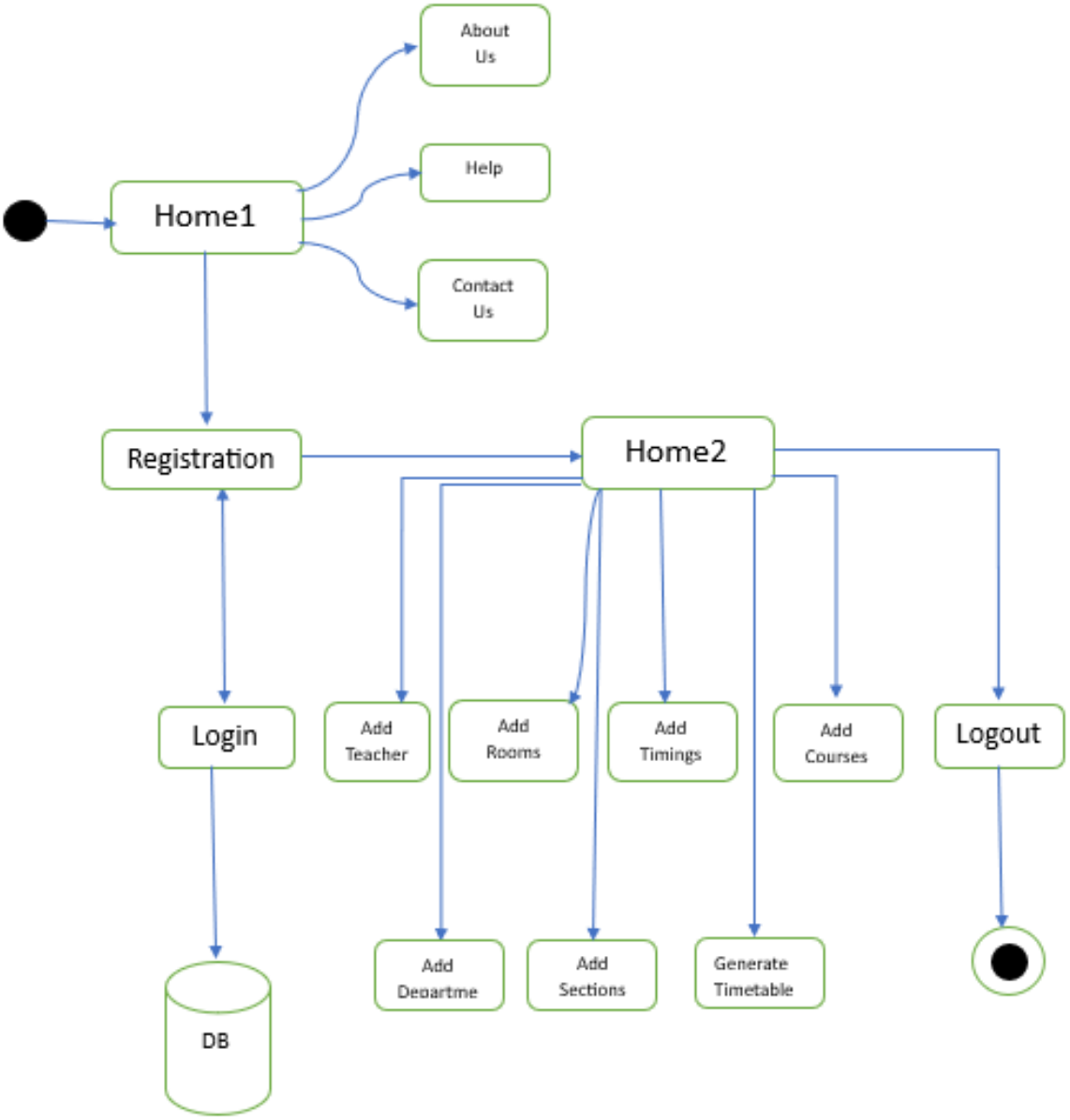


Fig 4.1 Architecture of Timetable Generation Application

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

- **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.

- **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.

- **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.

- **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.

- **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).

- **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.

- **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.

- **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.

- **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. It should be feasible for users to make manual corrections.

CHAPTER-6

SYSTEM DESIGN & IMPLEMENTATION

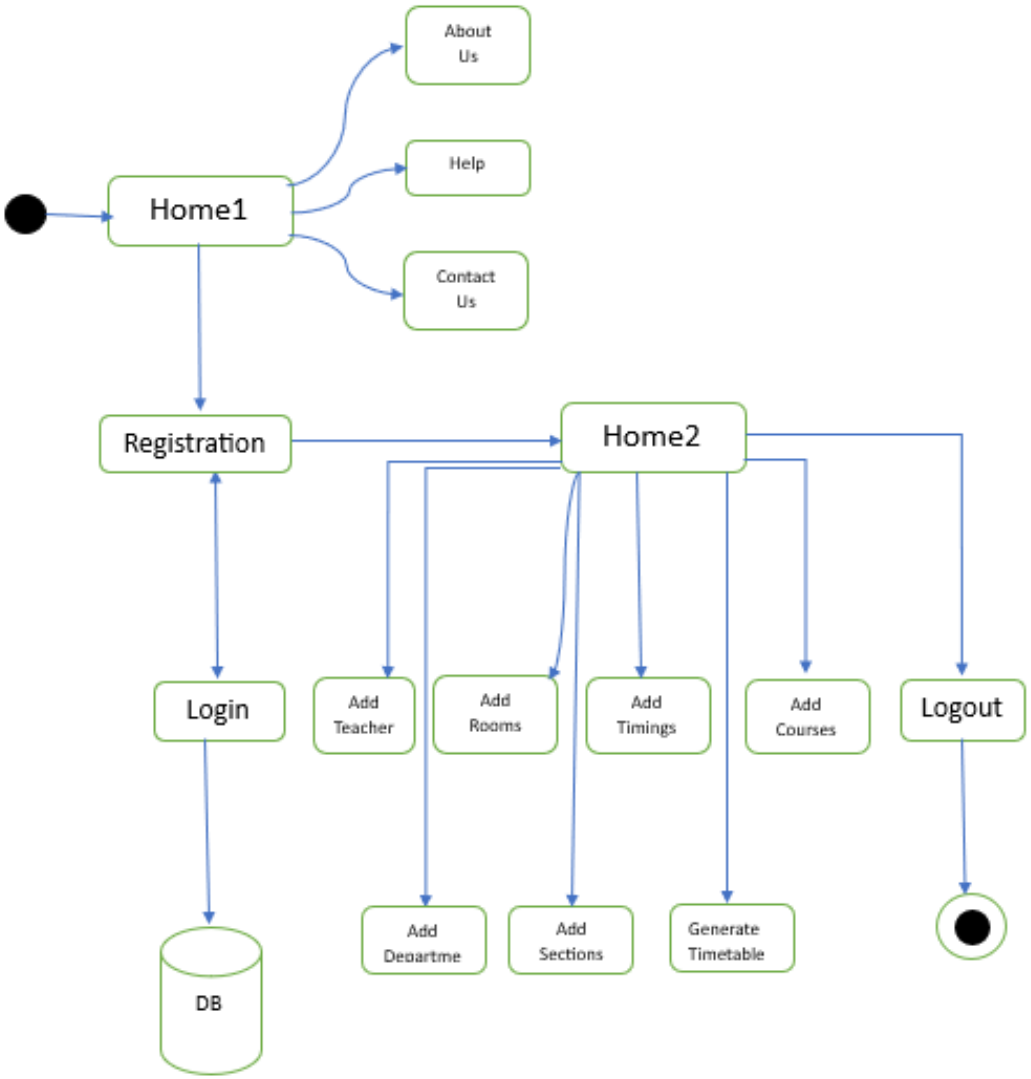


Fig 6.1: Architecture of Timetable Generation Application

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

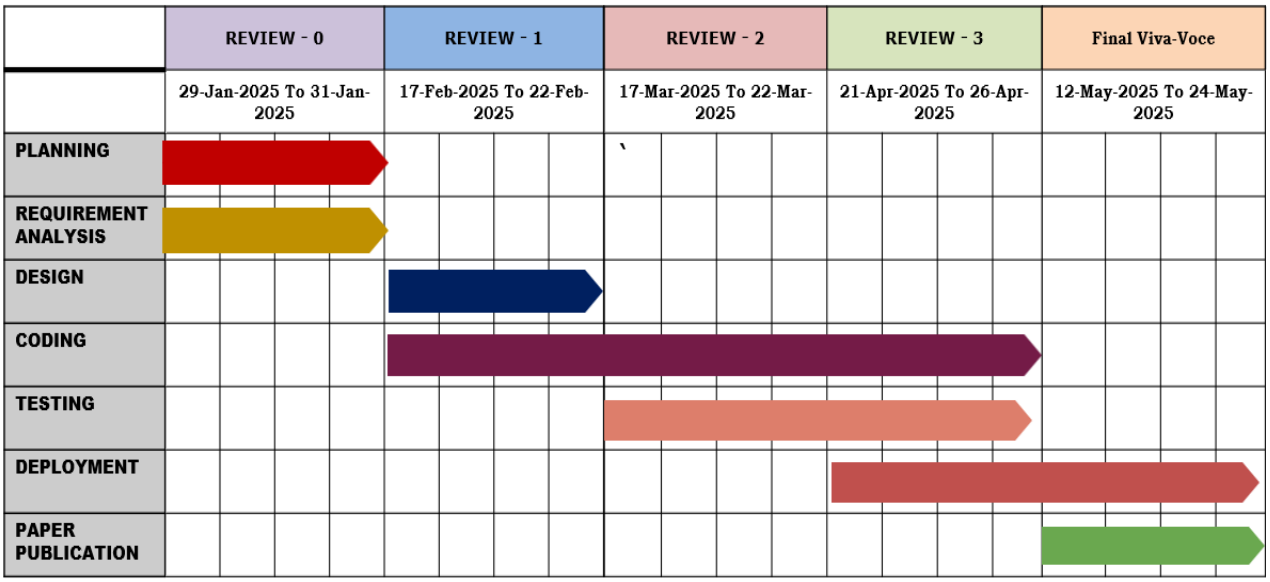


FIG.7.1 Timeline of The Project

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.

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
- Stress reduction through automatic scheduling
- The institution gains credibility through an organized and effective approach.

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.

9.1. Results

- **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. 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.

- **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

- **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.

➢ **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.

➢ **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.

➢ **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.

9.2. Discussion

- **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.

- **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

- **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

- **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.

- **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.

- **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.

- **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.

- **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 work 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]

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 username 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

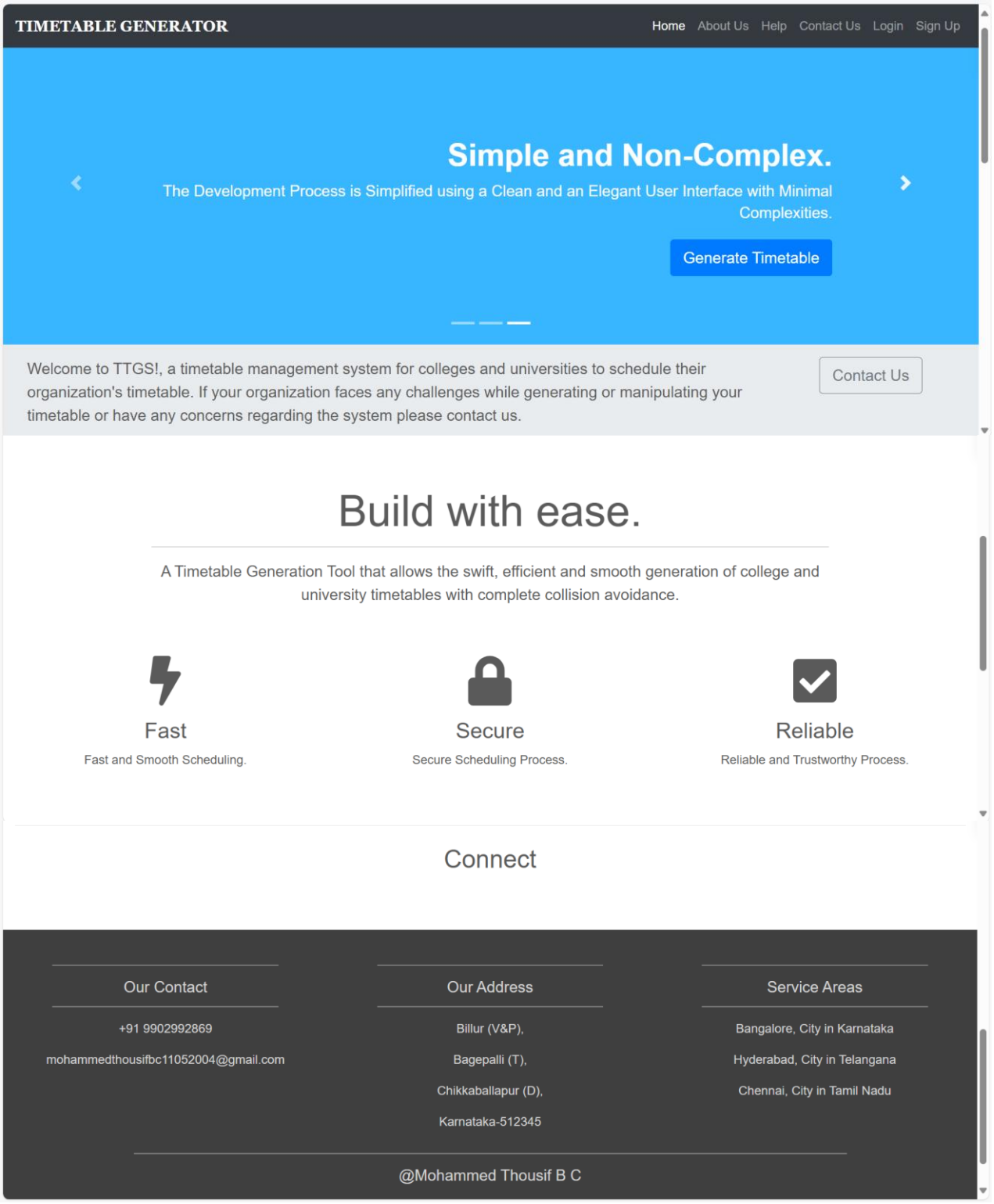


Fig.1.Home1 Page

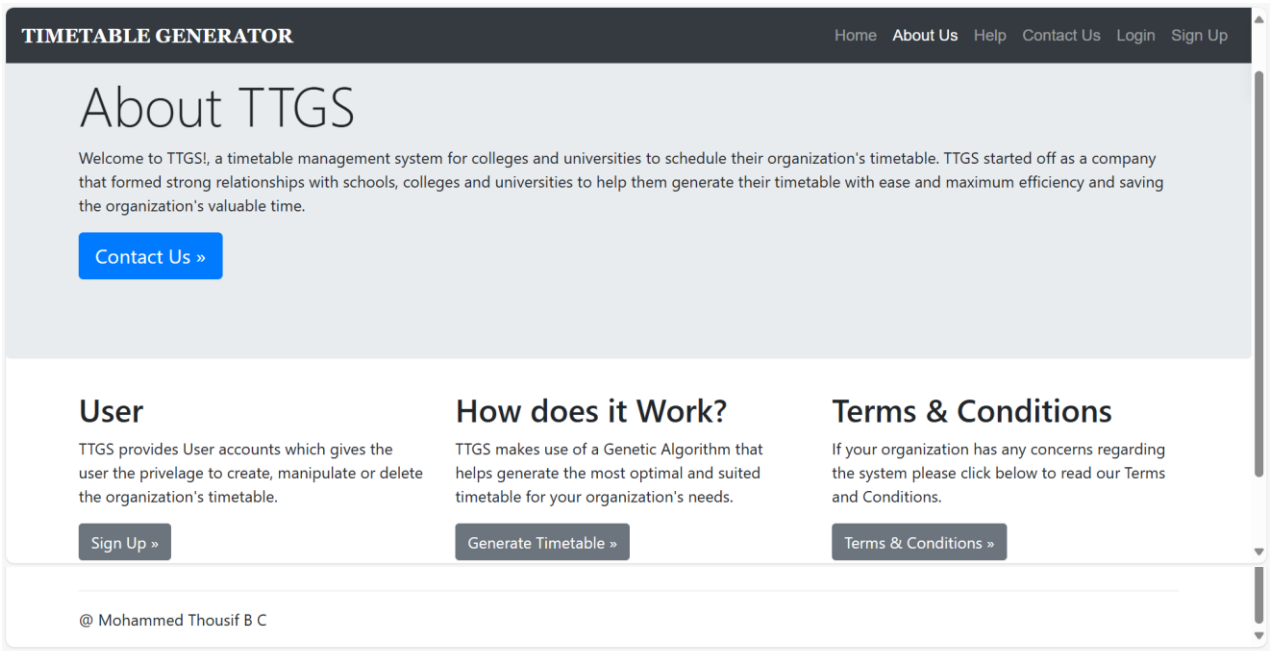


Fig.2.About Us Page

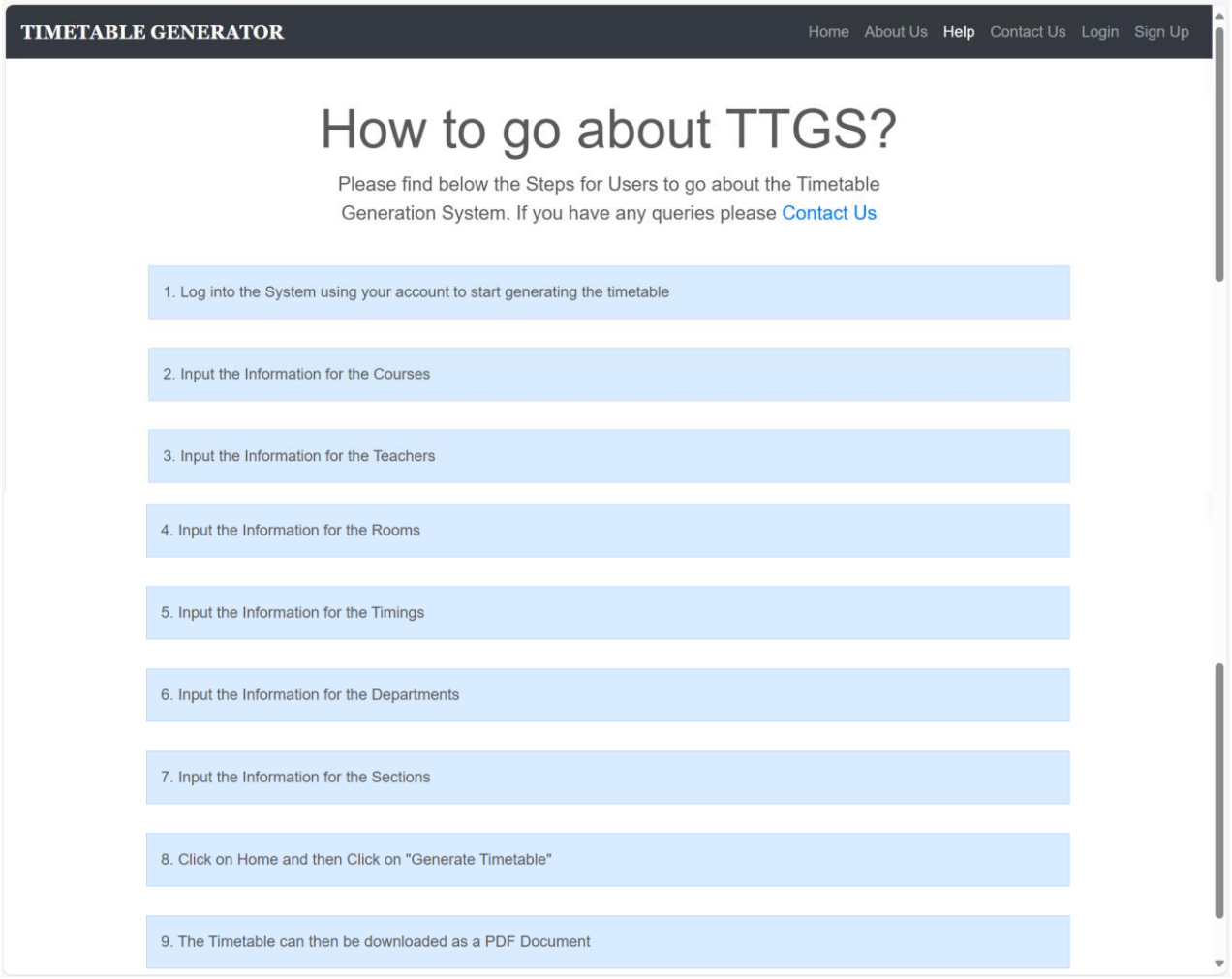


Fig.3.Help Page

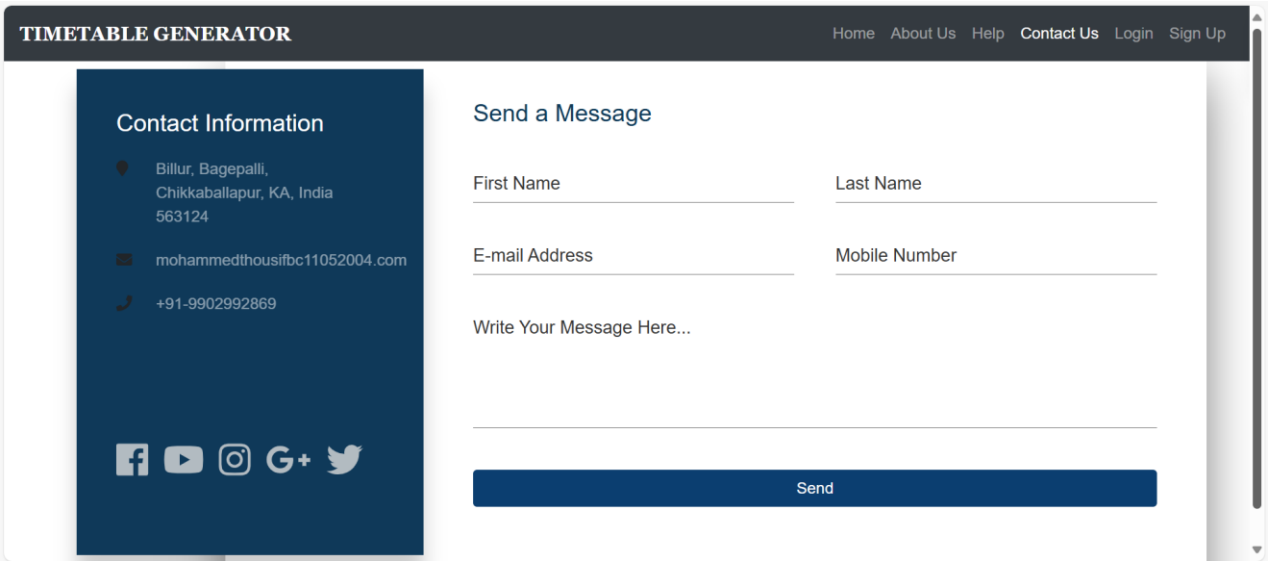


Fig.4.Contact Us Page

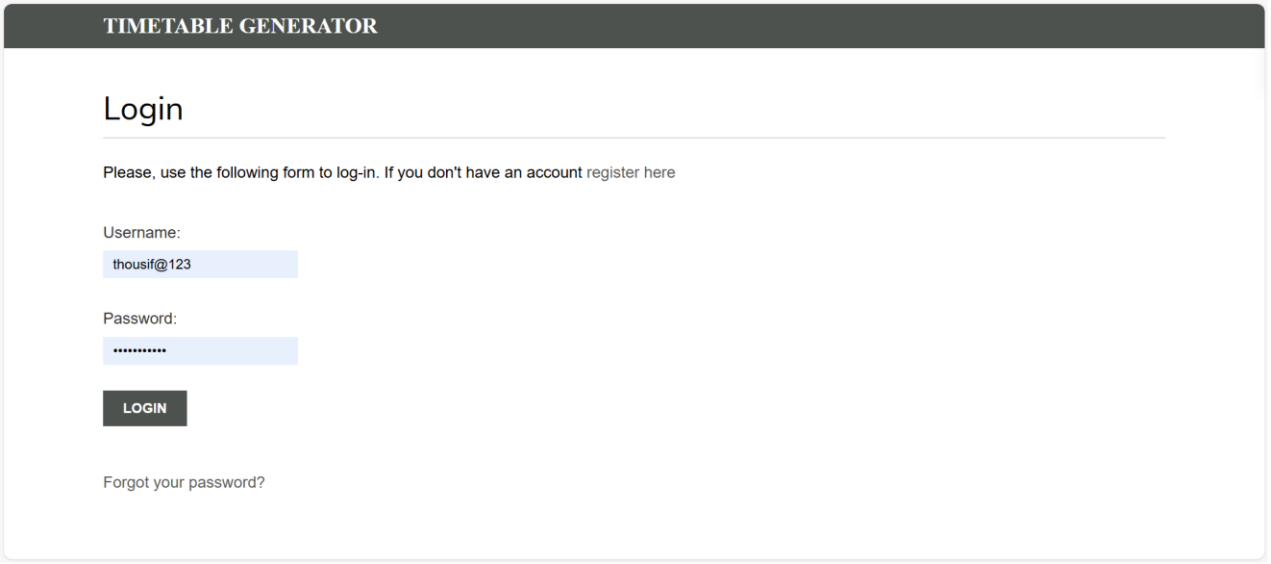


Fig.5.login Page

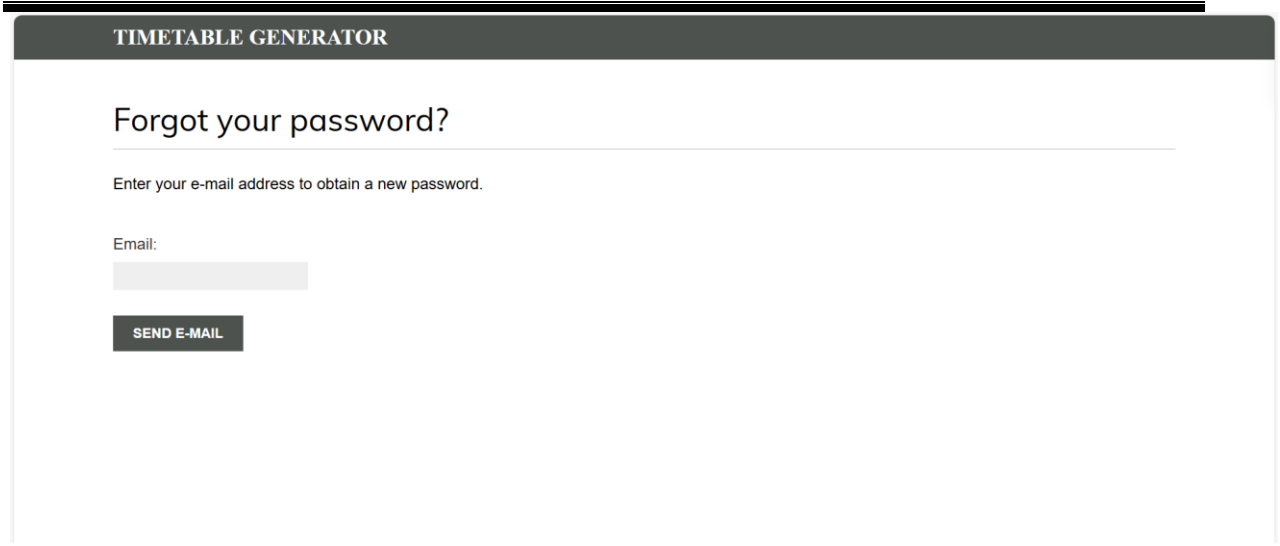


Fig.6.Forgot password Page

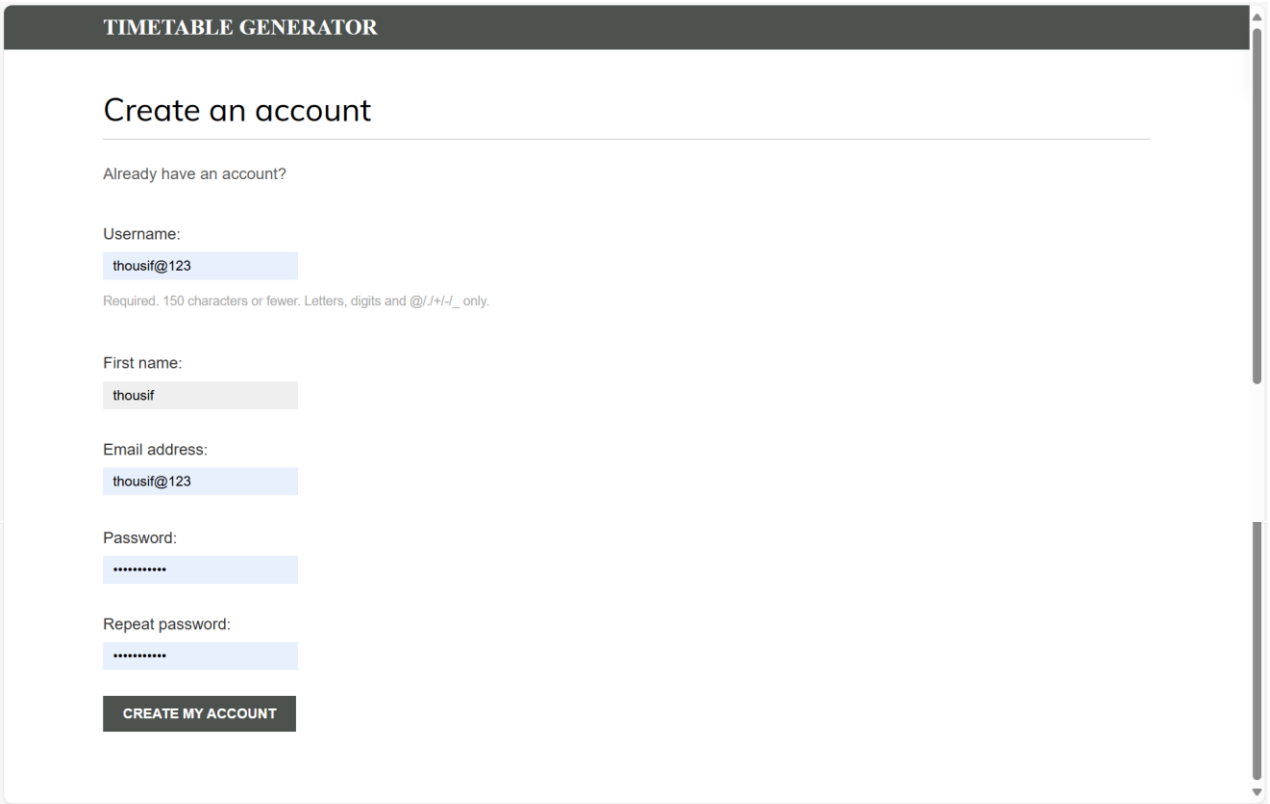


Fig.7.SignUp Page

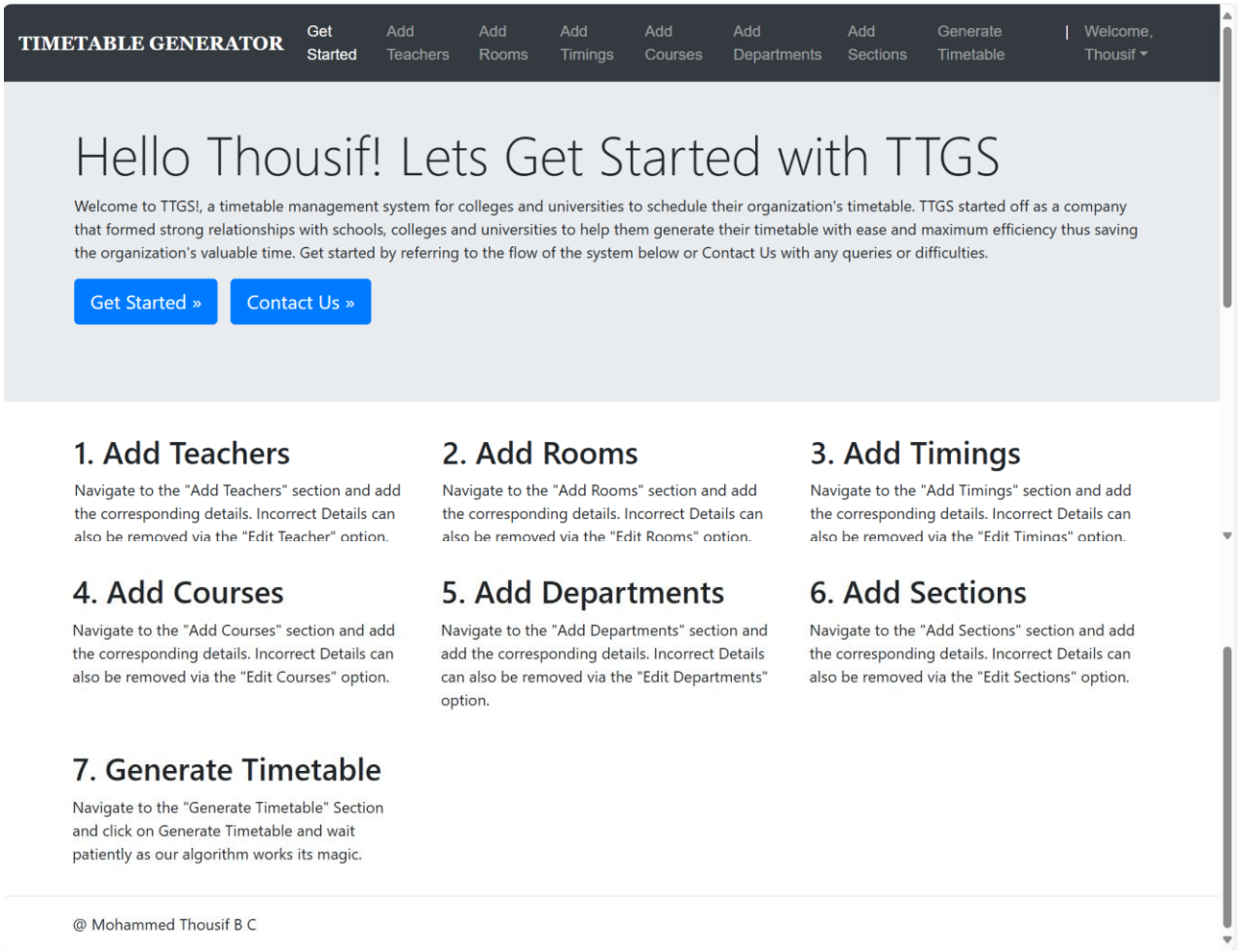


Fig.8.Home2 Page

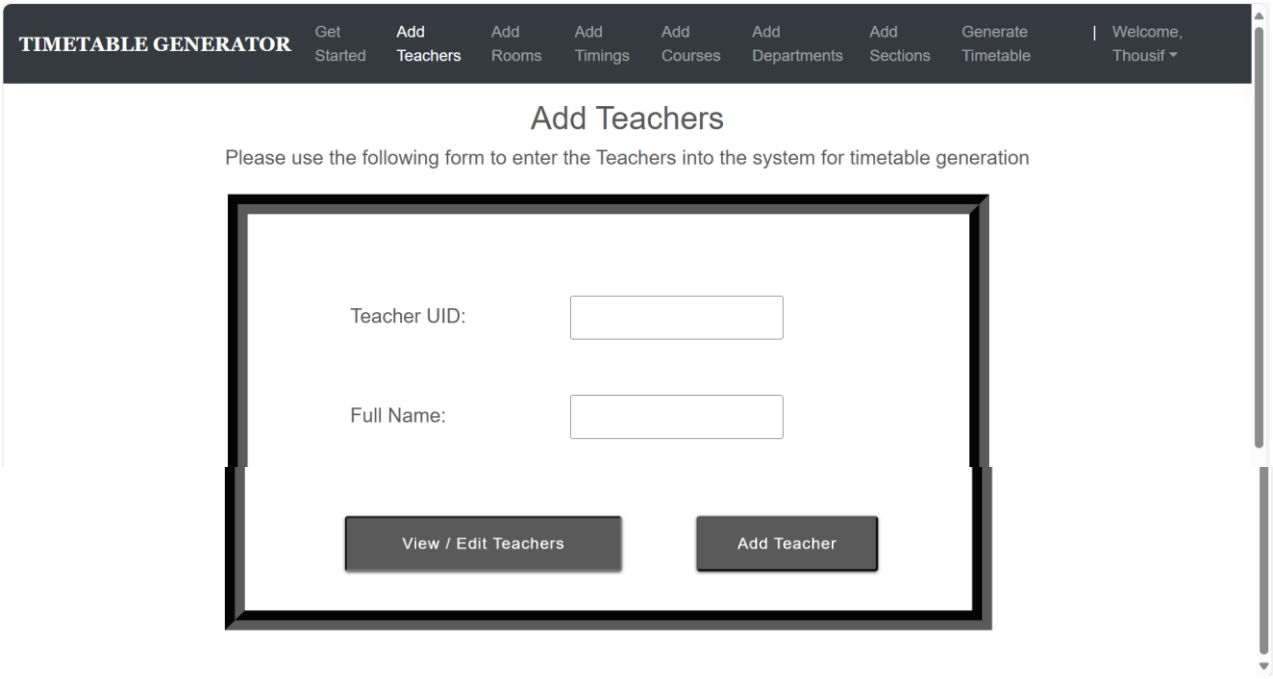


Fig.9.Add Teachers Page

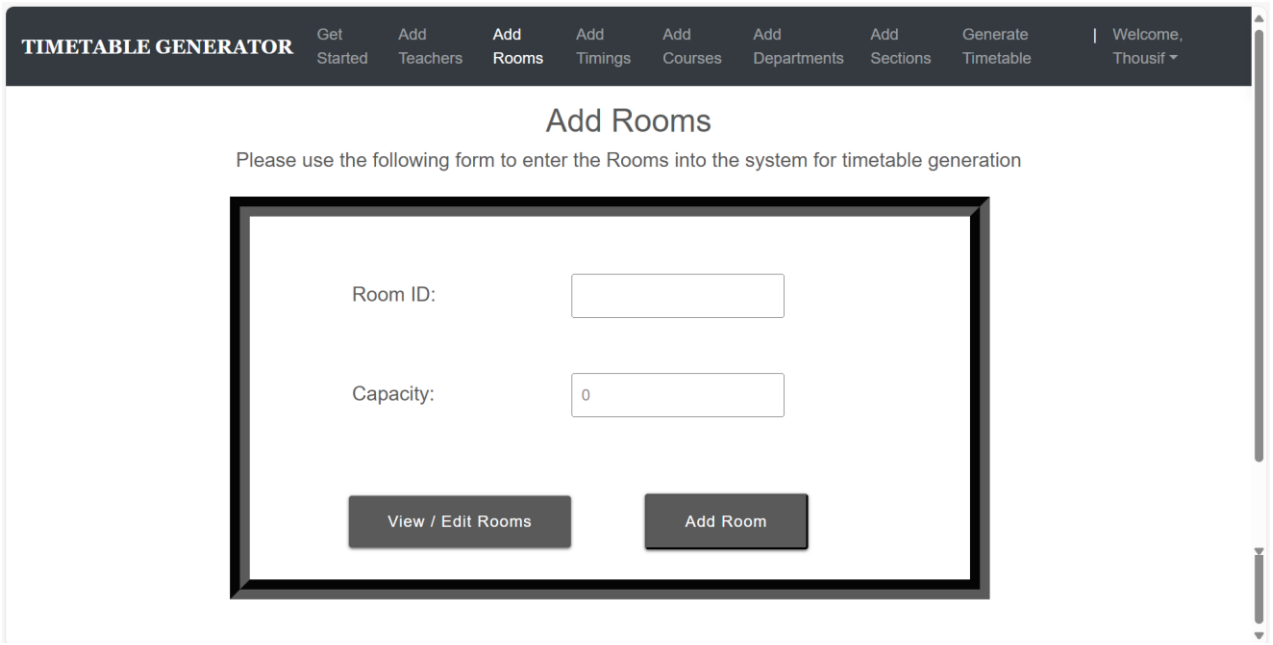


Fig.10.Add Rooms Page

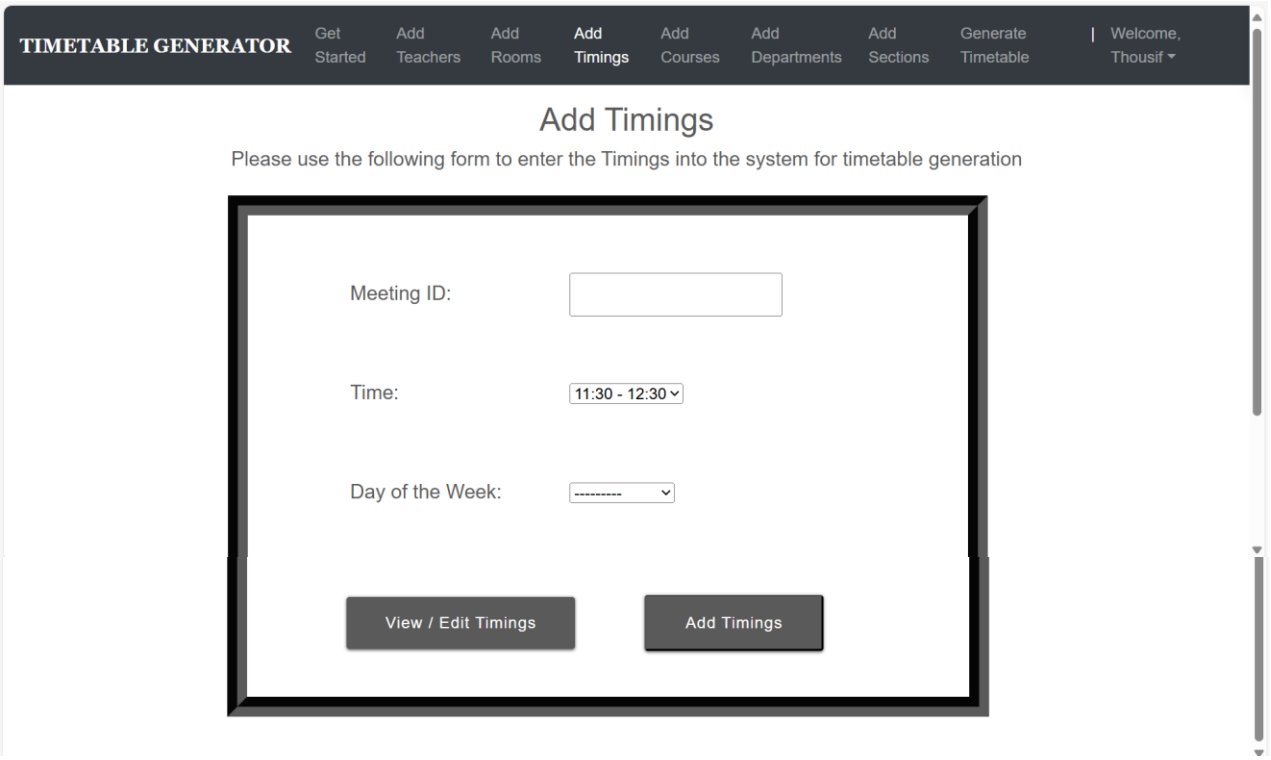


Fig.11.Add Timings Page

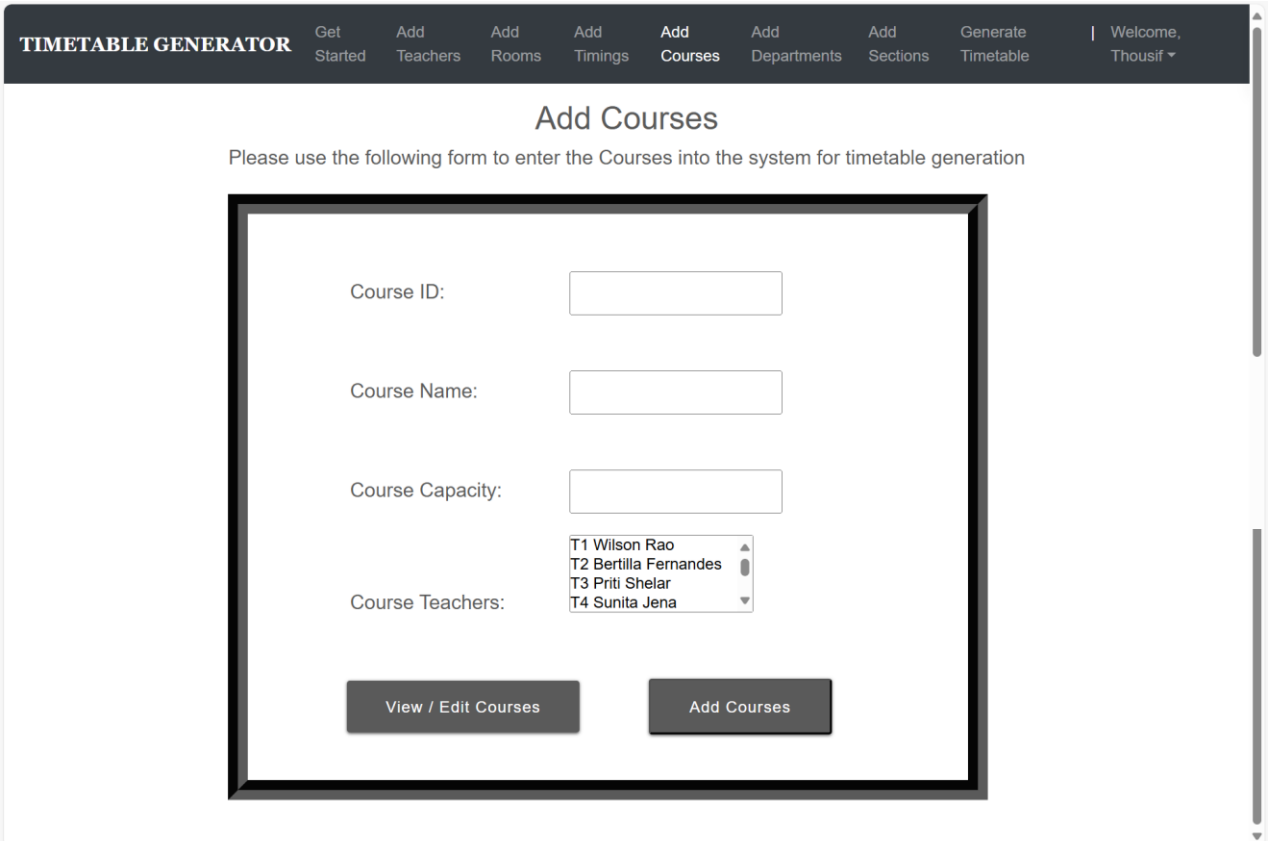


Fig.12.Add Courses Page

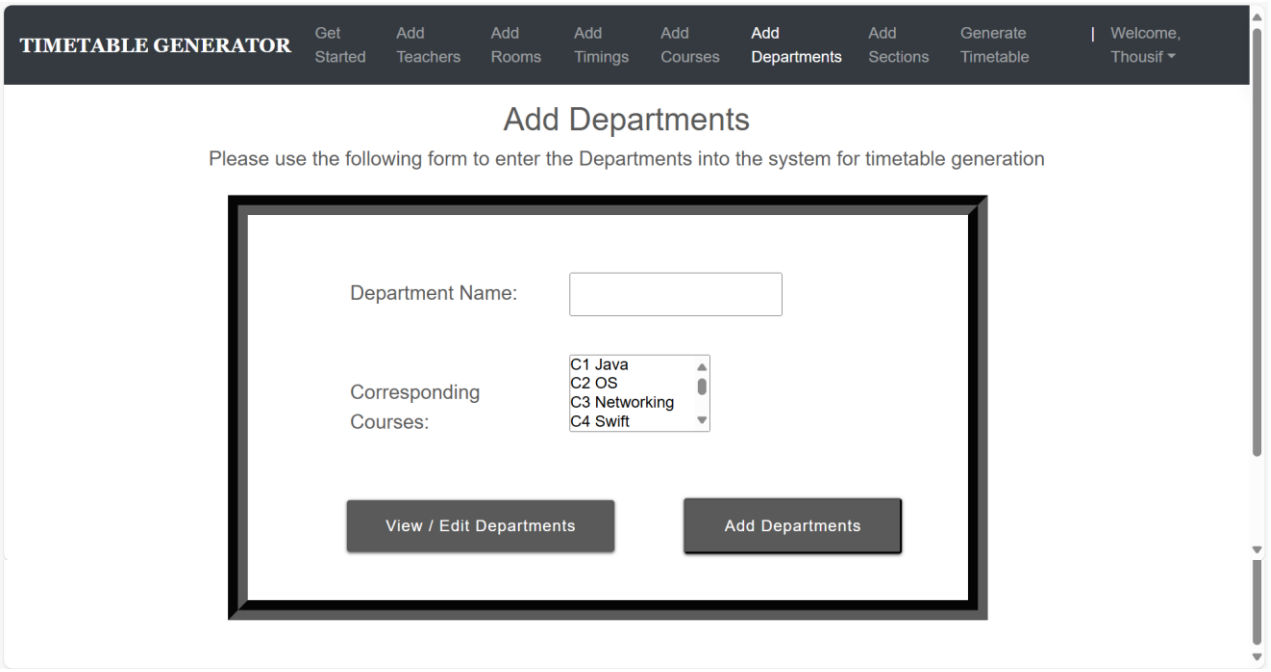


Fig.13.Add Departments Page

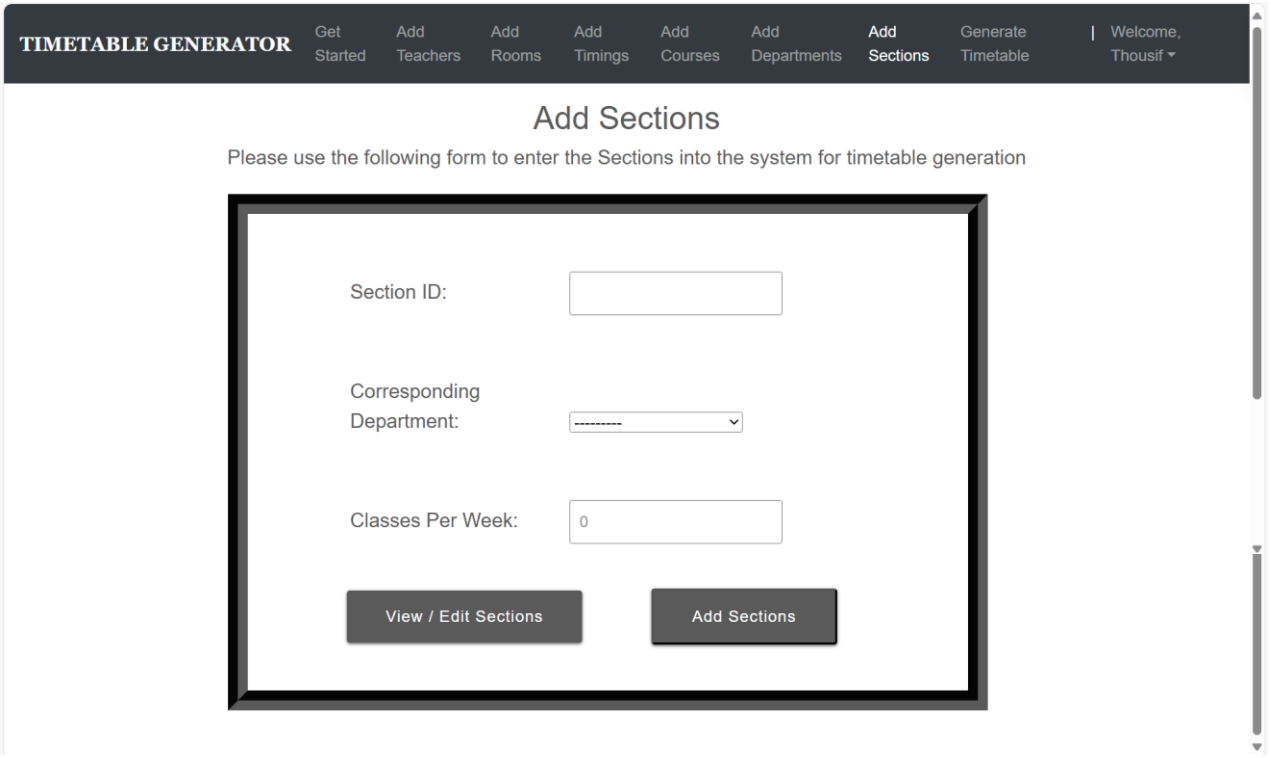


Fig.14.Add Sections Page

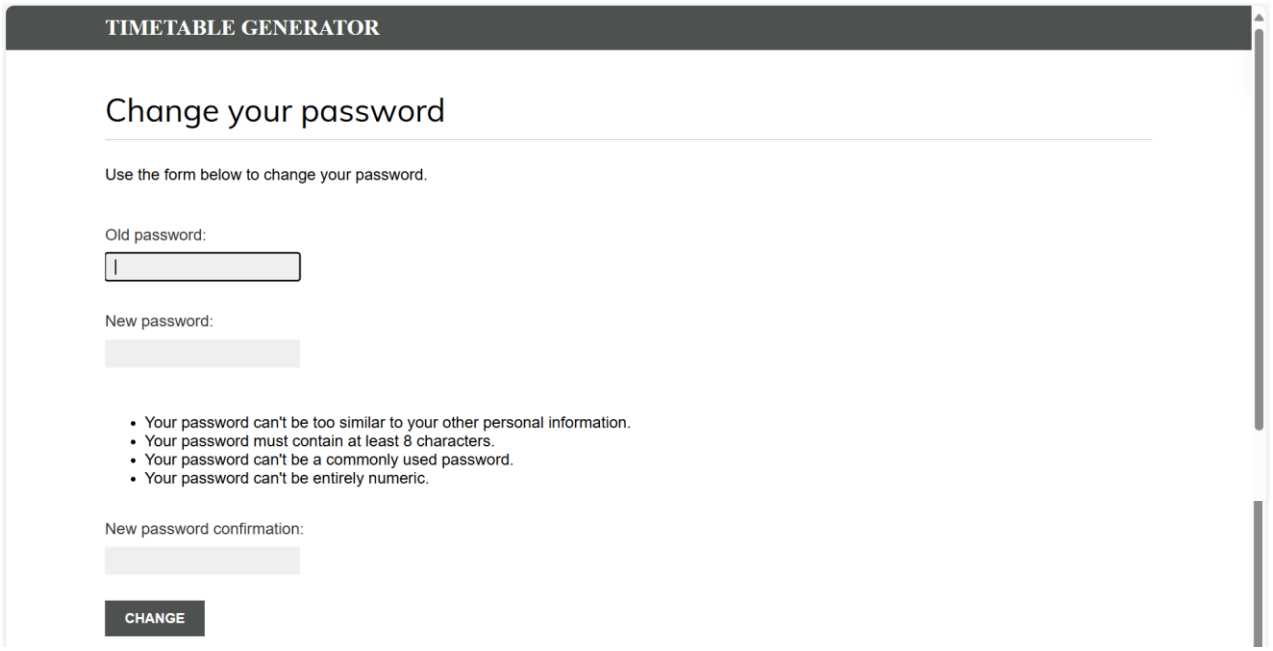


Fig.15.Change Password Page

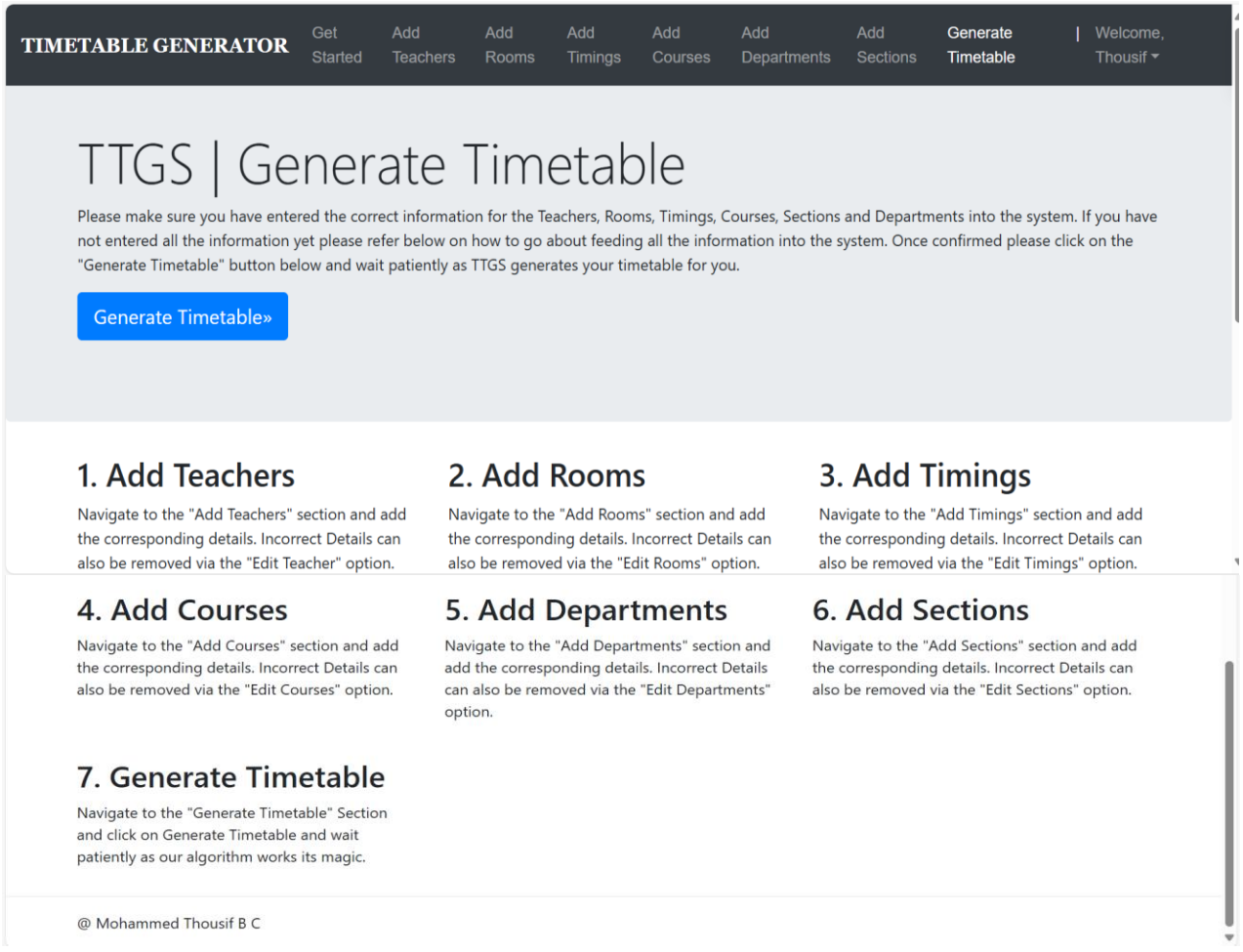


Fig.16.Generate Timetable Page

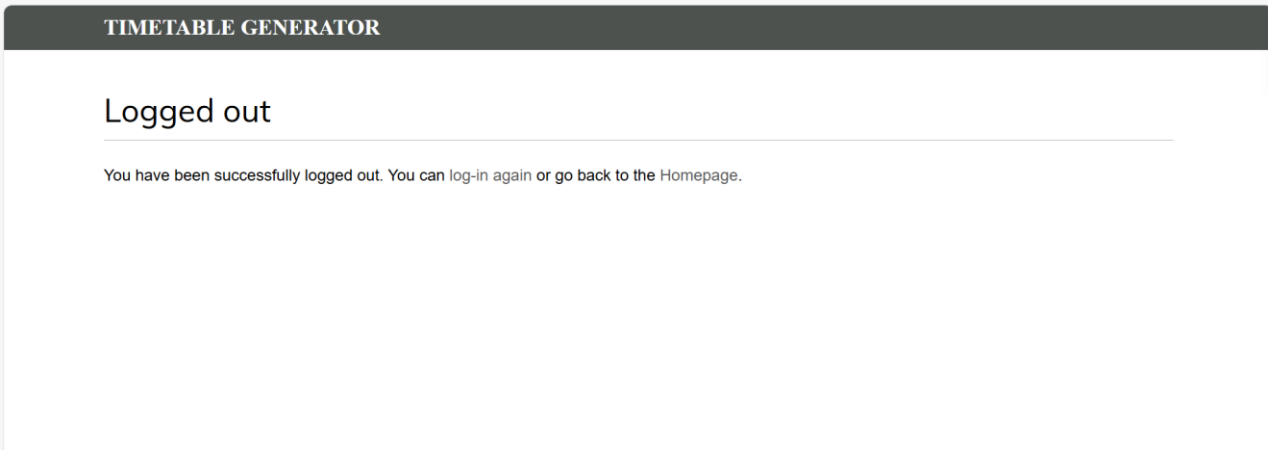
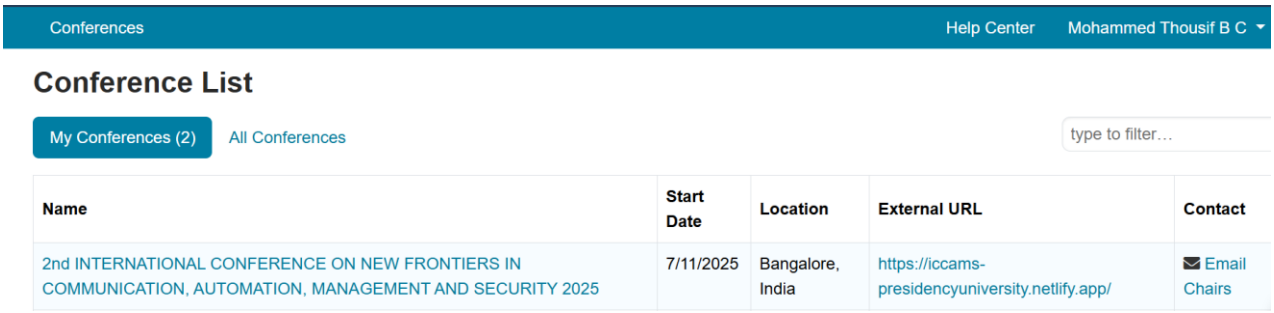


Fig.17.Logged out Page

APPENDIX-C

ENCLOSURES

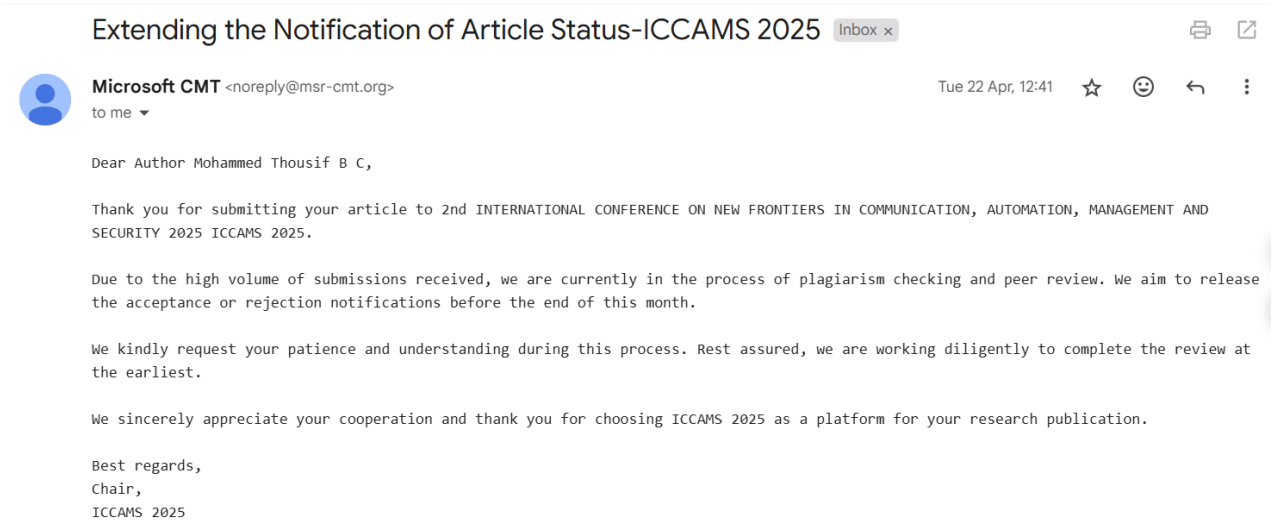


Conferences | Help Center | Mohammed Thousif B C

Conference List

My Conferences (2) | All Conferences | type to filter…

Name	Start Date	Location	External URL	Contact
2nd INTERNATIONAL CONFERENCE ON NEW FRONTIERS IN COMMUNICATION, AUTOMATION, MANAGEMENT AND SECURITY 2025	7/11/2025	Bangalore, India	https://iccams-presidencyuniversity.netlify.app/	Email Chairs



Extending the Notification of Article Status-ICCAMS 2025 Inbox ×

Microsoft CMT <noreply@msr-cmt.org> Tue 22 Apr, 12:41

to me

Dear Author Mohammed Thousif B C,

Thank you for submitting your article to 2nd INTERNATIONAL CONFERENCE ON NEW FRONTIERS IN COMMUNICATION, AUTOMATION, MANAGEMENT AND SECURITY 2025 ICCAMS 2025.

Due to the high volume of submissions received, we are currently in the process of plagiarism checking and peer review. We aim to release the acceptance or rejection notifications before the end of this month.

We kindly request your patience and understanding during this process. Rest assured, we are working diligently to complete the review at the earliest.

We sincerely appreciate your cooperation and thank you for choosing ICCAMS 2025 as a platform for your research publication.

Best regards,
Chair,
ICCAMS 2025

Sustainable Development Goals (SDGs).

Fig.18. An application to Generate Summer-Term Timetable Goals

Goal 4 aims to "ensure inclusive and equitable quality education and promote lifelong learning opportunities for all." Our timetable generator contributes to this by improving the efficiency and organization of educational institutions, allowing teachers and students to make better use of their time and resources. It minimizes scheduling conflicts, supports personalized learning paths, and promotes a smoother educational process, all of which enhance the quality and accessibility of education.

Mohammed_Zia_Ur_Rahman_Effortless_Timetable_Generation

by Mohammed Zia

Submission date: 27-Mar-2025 12:58PM (UTC+0530)
Submission ID: 2626630301
File name: Mohammed_Zia_Ur_Rahman_Effortless_Timetable_Generation.docx (3.69M)
Word count: 2584
Character count: 16529

Effortless Timetable Generation

Mohammed Thousif B C[1,*], Ullas H R[2], Md Ziaur Rahman[3]

[1]*UG Student SOCSE, Presidency University, Banglore-560 064, India, mohammedthousifbc11052004@gmail.com*
[2]*UG Student SOCSE, Presidency University, Banglore-560 064, India, ullasu213@gmail.com*
[3]*Assistant Professor SOCSE, Presidency University, Banglore-560 064, India, mdziaurrahaman@presidencyuniversity.in*

Abstract: The Timetable Generation is a software innovation designed to autonomously generate and manage an academic schedule without the inefficiencies, conflicts, or labour-intensive elements found with traditional timetabling processes. By using Python, Django, and SQLite3, the system is scalable and flexible enough for small or large institutions to guarantee real-time updates, conflict resolution, and optimized faculty workload. Its features include a friendly interface, through which faculty and administrators can have easy access, advanced filtering capability, and high data security in scheduling. More updates, which will include predictive analytics, AI optimization, cloud integration, and mobile accessibility, will further empower it to function as a tool of transformation in educational institutions all over the world.

1. Introduction

The Advanced timetable generation software automatically generates and updates academic timetables, replacing the inefficiency and conflict characteristic of traditional time-table management methods. It gives optimized timetables based on faculty availability, subject schedules, and institutional policies. The software is built in Python, Django, and SQLite3, giving secure data storage and real-time updates. Due to its high scalable architecture, it can function in schools and universities, adapting speedily to variations during peak demands.

The software offers a friendly user interface through which faculty and administrators can easily produce timetables without any technical knowledge. It ensures sophisticated filter options and resource allocation due to the removal of scheduling conflicts and better management of faculty workload while possessing full force emphasis on data security through encryption techniques and following the strictest data protection protocols for safeguarding sensitive information concerning academics.

It helps to simplify scheduling with the capacity to generate timetables automatically dependent on factors such as faculty availability, subject schedules, and institutional policies. It has a user-friendly interface that allows administrators and faculty to easily generate timetables or make adjustments without requiring technical know-how. This way, the system can handle the needs of small and large educational institutions, thereby adapting to the different environments and the unexpected updates or changes in real-time. The software optimizes the allocation of resources and avoids scheduling conflicts, therefore, improving efficiency and reducing administration staff burdens.

Going forward, the software will feature predictive analytics, AI-driven optimization, and cloud integration that would facilitate dynamic scheduling on faculty availability and student preferences. Upgrades include mobile access and integration with learning management and student information systems that can make scheduling much more efficient. This means the Automatic Timetable Generator would change the face of educational institutions all over the world. Real-time notifications and automatic conflict resolution will help maintain seamless operations. These capabilities will enhance decision-making, optimize resource usage, and minimize disruptions. In the end, the Automatic Timetable Generator will transform the way educational institutions manage their schedules worldwide, boosting efficiency and overall satisfaction.

2. Proposed System

The system is intended to effectively create and maintain a Summer Term Timetable through a systematic, database-based methodology. With HTML, CSS, and JavaScript on the frontend, it is intended to automate the scheduling process, making it accurate and user-friendly.HTML, CSS, and JavaScript are used for the frontend, while Django and SQLite3 are used for backend activities. The system provides accuracy, efficiency, and scalability. The system is developed to maximize the scheduling process for both students and staff, ensuring a seamless and user-friendly experience while being flexible and having real-time updates.

2.1. Major Modules and Features of the System:

1. User Interface Module:

This module provides a web-based, interactive interface for instructors and students to view and manage their schedules easily. System Components and Functionality

2. User Interface Module:

Implemented in HTML, CSS, and JavaScript, the module provides a responsive and user-friendly interface. Users are able to effectively search and filter schedules based on course, instructor, or time slot to provide smooth and easy navigation.

3. Input Processing Module

This module gathers necessary information, such as course information, instructor availability, and preferred time slots. It uses validation methods to ensure data integrity, which is accurate and consistent. It also detects and resolves scheduling conflicts by identifying overlapping time slots.

4. Timetable Generation Module:

Driven by Django, this module dynamically generates optimized timetables in real-time under predetermined constraints. It adjusts automatically to faculty availability and course load distribution, promoting efficient resource utilization and preventing scheduling conflicts.

5. Database Management Module:

Maintains course data storage and organization, faculty allocation, and time slot management with SQLite3. It provides effective retrieval, updating, and modification of data for uninterrupted timetable management. It maintains quick retrieval of data, update, and modification so administrators can easily make necessary schedule changes.

6. Output Module:

Presents the created timetable in an organized and user-friendly manner for easy reading. It also offers download or print schedule options for offline use. Real-time synchronization guarantees that any changes made

by users or administrators are immediately reflected within the system. Finally, the Error Handling and Logging Module detects and resolves scheduling conflicts, maintains the system's accuracy, and records database errors and system malfunctions for troubleshooting. This module enhances system stability by effectively managing invalid inputs, ensuring a robust and reliable timetable management experience.

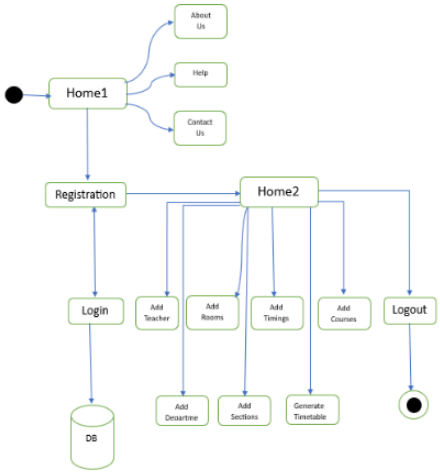


Fig.1. Architecture of Timetable Generation Application

2.2. System Overview:

Developed using Django and SQLite3 for backend processes, the system is guaranteed to maintain accuracy, efficiency, and scalability of timetable management. It avails a smooth scheduling process for students and instructors, with flexibility and real-time display.

7. Mathematical Model

The Timetable Generation System is composed of major steps that allow user interaction and generation of schedules to create an optimized timetable for students, administrators, and faculty. The system incorporates various modules, which are the user interface (UI), backend database, available resources (e.g., courses, rooms, and faculty), and conflict resolution schemes.

The major steps are:

1. User Registration and Authentication:

The system provides safe user authentication by authenticating and storing credentials like email, phone number, or social media logins in a secure database. Advanced security features such as token-based authentication and OAuth are used to protect user access.

Mathematical Representation:

User authentication is done by verifying credentials and security tokens to approve access.

User Login Validation=f (User Credentials, Security Tokens) →Access to Services

2. Course and Faculty Availability Input:

The faculty members show their availability, while administrators schedule courses and allocate room resources.

Mathematical Representation:

The system combines faculty availability, course schedules, and room assignments to provide accurate and efficient generation of timetables.

Course and Faculty Data=f(Faculty Availability, Course Timings, Room Resources)→Valid Input for Timetable Generation

3. Timetable Generation and Conflict Resolution:

Using constraints like course requirements, room capacity, and faculty availability, the system creates an optimized timetabling while successfully rounding off scheduling conflicts. Any scheduling conflicts or overlapping time slots are automatically resolved by conflict resolution algorithms.

Mathematical Representation:

Timetable Generation=f (Course Details, Faculty Availability, Room Availability, Course Load) →Generated Timetable without Conflicts

4. Conflict Resolution:

The system detects scheduling conflicts and uses resolution techniques to produce a finalized schedule.

Mathematical Representation:

Conflict Resolution=f (Detected Conflicts) →Resolved Timetable

5. Room and Resource Management:

The system organizes rooms according to course needs, capacity of the rooms, and faculty preference to make best use of spaces without allowing double booking.

Mathematical Representation:

Room Allocation=f (Room Size, Course Requirements, Room Availability) →Optimal Room Assignment

6. Real-Time Updates and Notifications:

Changes in the schedule, like changes in faculty or room, are updated in real-time throughout the system. Alerts to users are given for the changes so that all are kept in the loop immediately.

Mathematical Representation:

Real-Time Update=f (Schedule Changes, User Preferences) →Updated Timetable with Notifications

7. Data Security and Privacy:

User information, such as scheduling data, is stored securely and safeguarded in accordance with privacy policies. The users have the ability to log out, and their individual data is either securely deleted or stored according to their own desires.

Mathematical Representation:

Data Protection=f (User Data, Privacy Settings) →Secure Data Storage and Deletion.

8. Result Analysis

1. Home1 (Landing Page) & Navigation

The system begins with Home1, the first landing page. It offers access to:

About Us – System general information.

Help – User support and guidance.

Contact Us – Contact options for inquiries.

From Home1, users can go to the Login page to access system features.

Analysis:

A clear and organized landing page enhances user experience.

Direct access to assistance and support guarantees hassle-free onboarding for new users.

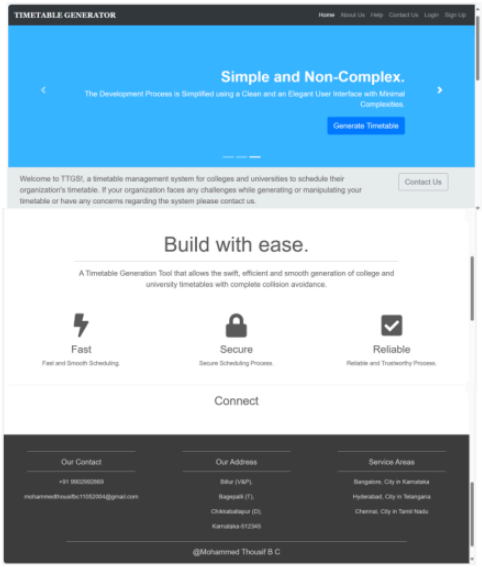


Fig.2 - Home1 Page

2. Login Process

Users (students, administrators, or faculty members) need to log in via a safe authentication page. The information is saved in the database (DB) for later retrieval.

Analysis:

Simplifying the registration process maximizes user interaction.

Effective data storage enables untroubled system operation.

3. Home2 (Main Dashboard)

Users are rerouted to Home2, the dashboard that offers access to core modules like timetable management, scheduling, and log out, upon login.

Analysis:

A well-crafted dashboard improves navigation.

Single-click access to all functionality saves time.

Module-wise Breakdown (Based on Diagram)

a. Add Teacher Module

Enables administrators to sign up faculty members.

Stores faculty availability and schedules for timetabling purposes.

Analysis:

Guarantees timely faculty scheduling.

Avoids workload imbalance.

b. Add Rooms Module

Facilitates administrators to specify available rooms for scheduling.

Contains room capacity and location information.

Analysis:

Maximizes space utilization.

Avoids room allocation conflicts.

c. Add Timings Module

Facilitates input of available time slots for courses and faculty.

Used in timetable generation and conflict resolution.

Analysis:

Ensures efficient course allocation to available slots.

Reduces conflicts in scheduling.

d. Add Courses Module

Allows administrators to add course information (course name, faculty, duration, etc.).

Integrated within the scheduling system.

Analysis:

Offers a structured course scheduling strategy.

Ensures seamless integration with faculty and student timetables.

e. Add Department Module

Supports department-based categorization of courses.

Analysis:

Assists in departmental-wise timetable organization.

Supports faculty and student-specific scheduling.

f. Add Sections Module

Allows administrators to specify student sections for various courses.

Analysis:

Guarantees effective batch-wise distribution of the timetable.

Avoids faculty double-up in allocations.

g. Generate Timetable Module

Automatically produces a timetable as per faculty, room, and course constraints.

Handles conflicts as per predefined parameters.

Analysis:

Lessens the administrative burden.

Guarantees efficient, error-free, and timely scheduling.

h. Logout Module

Permits safe logout for the users from the system.

Analysis:

Shuts out intrusions to users' data by others.

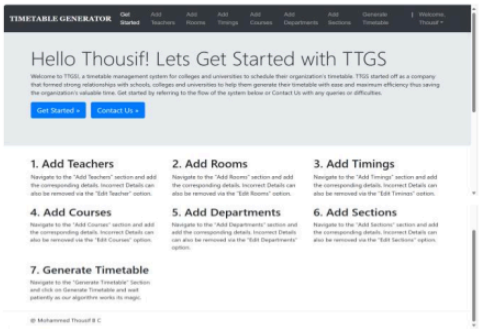


Fig.3 – Home2 Page

Advantages

Automated Scheduling – The system does away with the drudgery of manual scheduling through automated generation of the timetable, eliminating errors and saving time.

Enhanced Accuracy – Through the application of validation methods, the system makes sure that schedules are conflict-free and there is no overlap of time slots and inconsistencies.

Ease of Use – The interactive, web-based interface developed using HTML, CSS, and JavaScript gives an easy-to-use interface that enables students and teachers to easily view and manage schedules.

Real-Time Updates – All the updates done by users or administrators are updated in the system in real-time so that all the users see the current schedule.

Effective Resource Utilization – The system maximizes faculty availability, room reservations, and course assignments to make sure resources are used optimally.

Scalability and Flexibility – As it is based on Django and SQLite3, the system is scalable and adaptable to include additional courses, teachers, and students without affecting the performance.

Conflict Resolution – The system detects automatically and resolves automatically scheduling conflicts to ensure that two courses or two instructors are never assigned the same time slot.

Data Integrity and Security – The system maintains consistency, security, and retrievability of stored data through the use of a structured database.

Download and Print Support – The system supports downloadable and printable schedules to ensure that the user has access to his or her timetables offline for convenience.

Challenges

Complex Timetable Constraints – Handling a variety of constraints such as teacher availability, classroom constraints, and student preferences can complicate creating timetables.

Real-Time Synchronization Issues – Real-time synchronizing of updates for all the users in real-time may require heavy backend optimization and optimized database queries.

Data Validation and Integrity – Preventing redundant inputs, dealing with invalid inputs, and keeping data consistent across users can be complicated.

Scalability Issues – With the growing number of students, courses, and teachers, the system needs to handle large volumes of data without affecting the performance.

Dealing with Scheduling Conflicts – Even with automated conflict resolution, resolving constraints such as faculty availability issues or classroom shortages could still need administrator intervention.

Conclusion

The proposed work is an innovative solution for managing academic schedules. It automatically creates optimized timetables by factoring in constraints like faculty availability and subject schedules, which helps lessen administrative workload and enhance efficiency. Designed to be secure, scalable, and easy to use, it meets the needs of educational institutions of all sizes, from small schools to large universities. As more predictive analytics, AI-based optimization, and cloud integration enhancements come into being, the software will continue to optimize scheduling so much that its usage will be considered a mandatory part of an educational institution everywhere. Ultimately, Automatic Timetable Generator is about changing the game of academic scheduling in order to achieve a productive academic environment.

Appendix A. Pseudocode

```
Start
Navigation to Home1
    Options to be displayed: About Us, Help, Contact Us, Login
    If User chooses "About Us":
        Show system information and what it is for
    If User chooses "Help":
Display help page or FAQs
    If User selects "Contact Us":
        Display contact details for technical support
    If User selects "Login":
Go to Login Page

Login Process
    Accept username and password
    Authenticate credentials with database
    If valid:
        Redirect to Home2
    Else:
        Show "Invalid credentials"

Registration Process
    Take registration information (name, email, password, etc.)
    Save the user's registration details to the database.
    Show a message: "Registration Successful".

Go to Home2 (After Login)
    Show Options
    Add Teacher
    Add Rooms
    Add Timings
  Add Courses
  Add Department
  Add Sections
  Generate Timetable
Functionalities on User Selection:
```

If "Add Teacher" is selected:
Take teacher details, such as ID, Name, and Department.
Verify the Information Entered
Check the information entered.
Save the information of teachers in the database.
If "Add Rooms" is chosen
Capture the information of room details.
Validate the information entered.
Save all the details of the room in the database.
If "Add Timings" is chosen
Capture the details such as Start Time, End Time, and Session Type.
Validate the timing information
Insert the details of timing in the database
If "Add Courses" is chosen:
Obtain course details such as Course ID, Name, Semester.
Validate the input
Save course details in the database
If "Add Department" is chosen:
Obtain information such as Department ID, Name..
Validate the Details
Verify that all the information given is correct.
Save the department details into the database.
If "Add Sections" is selected,
Get the section details. It includes Section ID and Course ID
Verify the entered details
Save section details into the database.

Generation of Timetable
If the user selects "Generate Timetable",
Get teacher, room, course, and timing details from the database
While considering availability and scheduling constraints, generate a timetable

Logout
If the user clicks "Logout",
Destroy the session.
Take the user to Home1 (Login Page).

END

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

Mohammed_Zia_Ur_Rahman_Effortless_Timetable_Generation

ORIGINALITY REPORT

1%	1%	0%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Upper Iowa University Student Paper	1%
2	www.jmir.org Internet Source	1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		