EXAMINATION TIME TABLE GENERTOR

Mr. Mohamed Shakir [1], Mr. Raghu veer V[2], Mr. K Chakravarthi [3], Mr. Shreyas A Shetty [4] , Mr. Nikhil Raj [5]Mr. Mohamed Rashid [6]

[1]Assistant Professor, Presidency University, Bengaluru
[2] Undergraduate Student, Presidency University, Bengaluru
[3] Undergraduate Student, Presidency University, Bengaluru
[4] Undergraduate Student, Presidency University, Bengaluru
[5] Undergraduate Student, Presidency University, Bengaluru
[6] Undergraduate Student, Presidency University, Bengaluru

---***---

Abstract - Actually, the scheduling of examination timetables is one of the most important administrative challenges that educational institutions face. In examination timetables, the challenges of the same problem are encountered, and through the CSP algorithm Examination Timetable Generator addresses this very challenge by way of an automation and optimization means. It stands as a sophisticated mathematical model where the problem and its solution that involves scheduling a set of an examination can mostly be divided into three main component parts: it includes variables like exams scheduled for, the domains include time slots for all that a sequence of exams proposed to be held, and thus, the domain of constraints or application of constraints takes care of accomplishing all possible requirements.

***Key Words*:** Examination Timetable Scheduling ,Educational Institutions Administrative ,Challenges Automation Optimization, CSP Algorithm.

1.INTRODUCTION*)*

Exams are designed to test the knowledge or skill level of an individual or a group. The process that involves determining what is to be included in the exam, when it will be conducted, and how many questions will make up the exam, or the time limit for the exam, is known as examination timetabling. Most universities also require students to take final-year exams. Exams or tests that have not yet been scheduled can annoy or frustrate many people. Exams are usually conducted based on the difficulty level of the subject. The notion is that the most difficult paper will be given on the first day, allowing a limited amount of time for students to rest. Consequently, scheduling must be appropriate, and designing a schedule has a significant impact on the time and labour involved.

The primary aim of the project is to create a timetable generator to assist college and university faculty in producing a timetable. This capacity is anticipated to help improve and configure an optimal exam schedule according to their requirements. Evaluating a massive quantity of data can be time consuming and tiresome. By contrast, programmable schedulers can drastically reduce time and labour. A timetable generator is a tool designed using front-end and back-end tools. Timetable scheduling is a separate and complex subject that may encompass restricted room testing based on accessibility and room capability. There are numerous other constraints, including a minimum time slot and a maximum time slot based on the exam date and allotment of breaks for a few minutes. Furthermore, the schedule can also be generated under various conditions, including room reservation, writing rooms for some weak or physically challenged students, or any other option. The examination timetable generator, which is the core of our system, was built to assist institutions in preparing the exam timetable. Working on the outcomes of various studies, the institution notifies that a confidential routing exam timetable has been prepared.

Literature Review

Existing systems for examination timetables focus on different aspects, and each has some limitations. The limitations provided in most of the research clearly indicate that sufficient priority is not given to fulfilling practical requirements before conducting the implementation. The examination timetable generation was not addressed according to the requirements provided by this research. Therefore, improvements in the creation of the examination timetable were made with reference to the findings of the literature review. Improvements refer to the choice of technology, method, and ingredients, so they can complement each other from the aspect of difficulty.

In this study, the new web-based examination timetable generation is based on a literature review. Some technologies and methodologies used in related journals have been described. Some of the frameworks, such as the Genetic Algorithm, Artificial Bee Colony, and Tabu Search, are presented in this paper; two algorithms are the hierarchical fuzzy expert system and modified NSGA-II. Different programming languages that use a prior tool for creating an examination timetable include Java and PHP. In this context, the use of React.js-based software is the first of its kind to create an examination timetable. The experimental platform consists of different tools, such as API testing, database testing, performance testing, and interface testing based on black box testing and white box testing. It consists of different tools that use a combination of these to test the application as per the requirement. The main purpose of this study is to develop a web-based application for creating an examination timetable that can produce a timetable in less time and to solve the existing examination timetable problem and large-scale input data.

In this section, the development of the examination timetable generator is based on the concepts and phases in existing related journals. Generally, the development of web-based examination timetable generators follows the concepts based on the development of a web-based examination timetable generator, and the development is approached by using literature references. The main stages in developing web-based examination timetable generators are as follows: knowing the characteristics of the problem, collecting data requirements and objectives of the examination timetable, inventory of software needs and IT choices, choosing the IT to be used, designing or developing a comprehensive system, and validating the system.

2. Existing Examination Time Table Generators

A previous study presented a review of multiple examination timetable generators developed in the past. The existing timetable generators were then classified into seven groups according to their functionalities and possibilities of use. The study showed that most of the tools used for scheduling are Internet-based and mostly for the free use of public applications. In contrast to these tools, the new examination time-table generator will comprise a flexible set of functionalities depending on the access privileges and level of permission of the users. Thus, the new examination schedule system will be capable of endowing testing bodies, schools, and colleges worldwide with fast examination schedule generation software tools that can be used to automate the cumbersome examination schedule generation process of their institutions by incorporating a publicly accessible application programming interface. Furthermore, it was evident that the existing work is limited to the presentation of only the software system in isolation without its concurrent integration into already available tools for the faster generation of examination time tables for students in uncontrolled environments. There is a lack of existing assessments of the extent to which existing timetable schedule generating system features are engaging and adequately addressing user needs. For example, an information system capable of generating an examination timetable and allowing students to be informed about exam schedules via text messaging was introduced by experimentation results for a college, and all this was done through a service for the SMS module. Additionally, a study also tested the application of a web-based examination timetable generator, which is considered “a missing puzzle” to fill large existing gaps. In summary, our long-term research aims to expand the use of existing examination timetable generators for the integration of an AI-based multi-agent system for adaptive collaboration among stakeholders, including students.

3. React.js for Frontend Development

React.js is the JavaScript library used to build user interfaces. React.js efficiently updates and renders perfect components when web application data change. With React, developers can create an interactive and dynamic user interfaces based on a component-based architecture. This enhances the speed and performance of applications. This framework loads very quickly, as developers can easily use server-side rendered pages on a virtual DOM. Reacts are known for their fast and efficient performance. Furthermore, React provides a monitoring option that can be used to debug an application. In addition, it excels in terms of reactive programming.

The Because the Examination Time Table Generator is web-based, technology to develop the frontend side is required. The timetable generator will have some pages that show the client-side data, such as the list of subjects, room lists, lecturers, and exam time slots. React is used to handle the right time framework. Act is designed according to a structured framework that focuses on the placement of documents, allowing for a more structured HTML page. In React, the application flow is delivered from the parent to the child component. React is lightweight but very powerful, allowing for the reuse of the UI component code, which can update and render efficiently.

The Some projects and case studies have already used the React framework to solve their problems. For example, one project used the React framework to build interactive data visualization. Another project used the React framework to design an attractive user interface. We also face challenges in deploying the act. The challenge of deploying React is to refresh only the main product page after inserting or adding a new product to the database. The framework has not been updated. For this reason, we need to use state management tools for this method. User interfaces (UI) are designed with the aim that user interactions with the application are carried out effectively. Therefore, in the modern era, there are many frameworks for building UIs. An example is the React Framework. With React, UIs can be designed more effectively and reactively. The page can also be divided to make it more manageable in the code. Act is a powerful framework for building UIs. It offers amazing tools and libraries for developing UIs for Web development. In the context of building educational applications such as this web-based timetable generator, web technology must be aware that it is important for generating user experience on the site. Therefore, it must be handled well in today's framework.

4. Python for Backend Development

Python is an advanced backend development language that many developers prefer. Python is straightforward, and its syntax is relatively close to that of the English language, so a beginner or an experienced developer will be much more comfortable focusing on machine learning, data analysis, and other AI-related topics. Hence, Python will be the best option to start in backend development. Python is a high-level programming language that is very popular for backend development because of its simplicity and lengthy association, which can be easily read by humans. For backend development, the main features of Python include object-oriented programming, high-level data structures, dynamic binding, dynamic typing, and idiomatic expression. The fundamental reasons for using Python for the backend development are as follows**.**

1. Extensive Libraries and Frameworks: Python libraries for data manipulation, calculative operations, and others for algorithm implementation. Python also has a variety of web development frameworks that are useful for developing REST APIs.

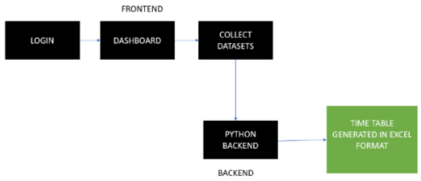
2. Backend Processing: Python relies on message brokers for backend processing so that we can start any backend test at any time and start a task later. These can be extended to real-time background checks and are effective for server-side programming.

3. Well-Documented Tools: Python libraries offer an easy-to-learn format. These include good documentation and many images to guide us. Thus, we can learn many new things by simply reading the Python documentation. With the right amount of resources, a development team can facilitate the establishment of a custom, rich REST API that can handle all internal connections of the frontend and the customer.

The current difficulty with Python is that one has to deal with database management issues and API monitoring issues. Python can be used to deploy the REST APIs across all connectors as effectively and efficiently as possible across all compatible applications. Python is, beyond reasonable doubt, the primary language for building apps. Various technologies make server-side programming simple and user-friendly. Therefore, the server-side development portion of the project was successfully completed using Python script.

5 Proposed Methodology

It would need to consist of three phases, plan, execute, and evaluate. Planning focuses basically on the analysis of the unit; hence it also incorporates information that needs to be gathered in preparing the system as a means to develop the timetable generator. In execution, a means to generate an examination timetable will have been developed. In order to develop the prototype, languages were to be used that could meet the demand requirements. Therefore, considering the server, Python's language



has been best estimated in terms of big data and CPU memory usage. The client's system has chosen to display modern and responsive views. The tools that were to aid in development were code editors.

Data sources: The following data sources would be used to gather data using a documentary technique. Documentary analysis was used to gather data for the study. Major source of data was obtained from textbooks and other sources while designing the system. Information obtained was converted by byte-code to meet the requirements of the study. Dataset: Imagery data was collected with the use of a variety of tools used to search for datasets. The relevant object file images used in testing the model were downloaded. The various phases of the development process of the system include the planning phase, designing phase, coding phase, and evaluation phase. It was carried out during the development process of the system by adopting the model of an iterative process. This model was adopted because of its ability to produce a better system. The adoption of the iterative process has helped to display the improvement in the system, as well as being able to embrace change and correction of mistakes. Iterative processes are carried out in cycles. These consist of two iterations of permutation. Repeating these cycles up to achieving optimized models culminated in producing a final phase with the final system at hand.

6. Data Collection and Dataset Preparation

The data collected in the research work includes details of students, departments, exam time slots, a sequence of courses for each level of student, and exam intervals from the 1st and 2nd semester exam timetables of the academic session. The exam time slots per day start from 08:00 AM. The interval of the time slot for the exam is 60 minutes from the morning to the evening of the day for different semester exams. All the time slots of the exam run up to the end of the day for the scheduling of the exam to take place. The data were collected from the exam and academic section of a department of a deemed technical institute. The data were available for examination periods in December/January and April/May. The sources of data are available at a convenient time and are recorded by the staff in the institute.

7 Results and Discussions

This project designed an examination timetable generator that has proven to reduce the time required for the production of a timetable and remove multiple sources of potential and hidden bias. As would be expected, it was shown to become more effective as the number of iterations of the genetic algorithm, as would be expected. Shortening the penalty for an increased degree distance may also speed up the examination scheduling. The project results are achieved using React.js for designing a frontend and Python for a backend requiring more space in the library storage, which can be improved in every test of the test administration. The relevance of this project can either provide an enhanced user experience or provide a series that follows up future implementations in an educational setting.

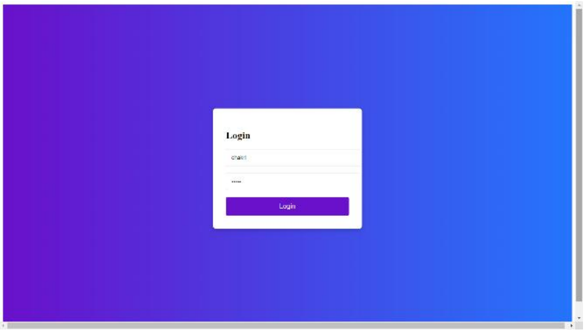


Figure 1: Login page

Figure 1: The Login page displays a 2 input fields to collect username and password credentials .

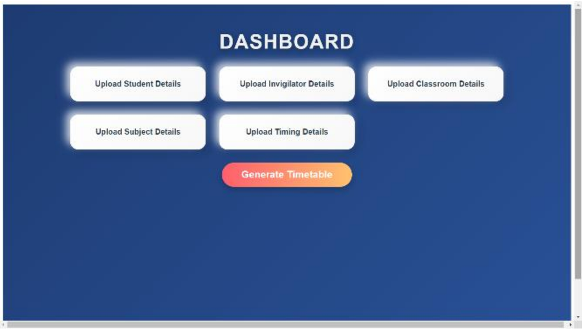


Figure .2: Dashboard Screen

Figure 2: The Dashboard screen provides the ability to choose the required datasets for the time table generation.

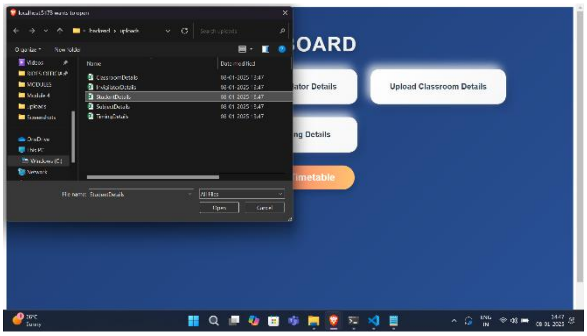
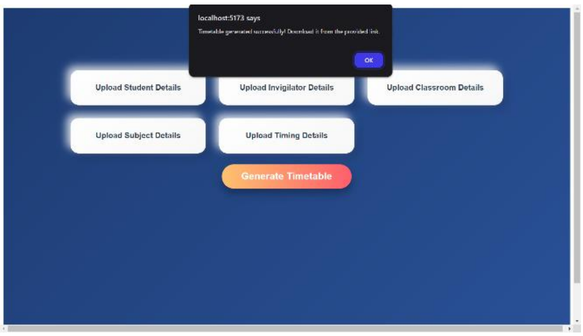


Figure 3: Choosing the Dataset

Figure 3: Choosing the Required Datasets for generating Examination time table .


localhost:5173 says

Timetable generated successfully! Download it from the provided link.

Figure 4: Time Table generated successfully

Figure 4: After collecting every dataset required and when generate button is clicked we will prompted with the successful time table generation

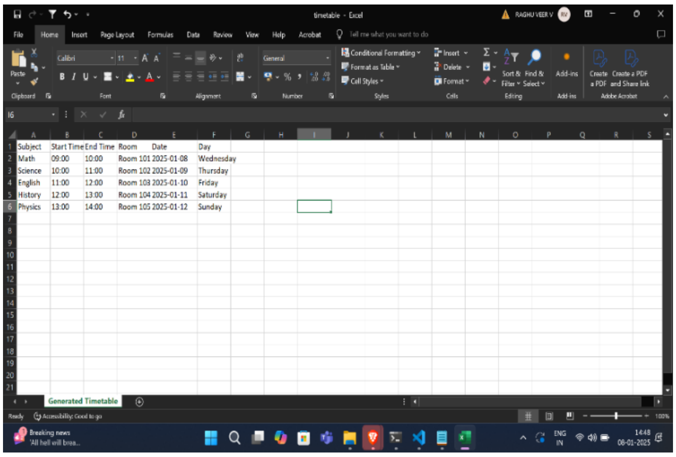

Subject	Start Time	End Time	Room	Date	Day
Math	09:00	10:00	Room 101	2025-01-08	Wednesday
Science	10:00	11:00	Room 102	2025-01-09	Thursday
English	11:00	12:00	Room 103	2025-01-10	Friday
History	12:00	13:00	Room 104	2025-01-11	Saturday
Physics	13:00	14:00	Room 105	2025-01-12	Sunday

Figure 5: Generated Time Table in Excel Format

Figure 5: After Generating the time table, we can find the excel file of time table in the backend folder and file namely timetable.xlsx.

The image depicts a user interface (UI) for an examination timetable generator. The UI is clean and modern, with a dark blue background and well-spaced buttons. The primary button is a large, vibrant "Generate Timetable" button, suggesting the main functionality of the application.

Above the "Generate Timetable" button, there are five smaller buttons labeled "Upload Student Details," "Upload Invigilator Details," "Upload Classroom Details," "Upload Subject Details," and "Upload Timing Details." These buttons likely allow users to input the necessary data for the timetable generation process.

The presence of an error message at the top indicates that there might be an issue with the system or the input data.

Overall, the UI appears intuitive and user-friendly, making it potentially easy for users to navigate and generate examination timetables efficiently.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to all those who contributed to the successful completion of

[1] this project. We are deeply indebted to [Name of Supervisor/Faculty], whose guidance, encouragement, and invaluable expertise have been instrumental throughout this endeavor. Their insightful feedback and unwavering support have been crucial in shaping the direction of our research and refining our approach. We would also like to thank [Name of Department/Institution] for providing us with the necessary resources and a conducive environment for conducting this research. Finally, we extend our heartfelt thanks to our families and their constant support and unwavering belief in our abilities.

Conclusion

The commencement of this development work gave birth to the application, which is an examination scheduler that aids schools, universities, and other institutions in conducting their schedules. This tool can be used to plan examination timetables for practical-based courses, course unit examinations, mid-term examinations, end-of-semester examinations, research projects or dissertation defences, and interview schedules. The application interfaces are web-based, and the implementation of the technology used is important to education, as it is based on writing and rolling out from specific journals that focus on these technologies aimed primarily at educationalists' consumption. Furthermore, this tool, which is being developed to automatically schedule examinations, is timely and worth undertaking.

References

[1] C. P. Gomes and C. M. R. de Souza, "A hybrid approach for university exam timetabling," in *Proceedings of the 2016 International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA)*, 2016, pp. 72–78, Doi: 10.1109/CIVEMSA.2016.16.

[2] F. Gendreau, G. Laporte, and M. Séguin, "A tabu search heuristic for the vehicle routing problem with time windows," *Journal of the Operational Research Society*, vol. 44, no. 5, pp. 482–490, 1993, Doi: 10.1057/jors.1993.61.

[3] M. R. K. M. K. De, "Exam timetabling using simulated annealing," *Journal of the Operational Research Society*, vol. 52, no. 11, pp. 1281–1289, 2001, Doi: 10.1057/palgrave.jors.2601291.

[4] S. S. R. Anwar and M. I. Sarfraz, "Optimization techniques for academic timetable generation using genetic algorithm," in *Proceedings of the 2017 International Conference on Advances in Computing, Communication and Control (ICAC3)*, 2017, pp. 49–56, Doi: 10.1109/ICAC3.2017.7896177.

[5] A. B. B. K. P. Bansal, "Time-table generation for educational institutions using genetic algorithm," *International Journal of Computer Applications*, vol. 72, no. 11, pp. 34–39, June 2013, Doi: 10.5120/12784-7312.

[6] A. T. P. G. B. V. R. S. K. A. Narayan, "A comparative study on optimization algorithms for exam timetabling," *International Journal of Computer Applications*, vol. 111, no. 2, pp. 1–6, February 2015, Doi: 10.5120/19735-4146.