**EUROPEAN UNIVERSITY OF LEFKE**

FACULTY OF ENGINEERING

Graduation Project 2

Timetable Generator

# Delight Ikechukwu Azuogu

# 194053

The project I chose to create is a timetable generator. The concept of the project is to create a web app that schedules the course timetable, i.e. it assigns a course, a class, and a time and a day in the week that the particular course will be taught.

This app can be used in school settings to eliminate the brainstorming that comes with creating a timetable for that semester because it checks for any free spaces in the timetable and assigns the course to any of the free spaces.

**Supervisor**

**Prof. FERHUN YORGANCIOĞLU**

**11th June 2023**

**Table Of Contents**

[**Delight Ikechukwu Azuogu 1**](#_heading=h.gjdgxs)

[**194053 1**](#_heading=h.30j0zll)

[**1. Introduction 2**](#_heading=h.3dy6vkm)

[1.1 Problem definition 2](#_heading=h.1t3h5sf)

[1.2 Goals 3](#_heading=h.4d34og8)

[**2. Literature Survey 4**](#_heading=h.2s8eyo1)

[**3. Background Information 4**](#_heading=h.17dp8vu)

[3.1 Required & Used software 4](#_heading=h.3rdcrjn)

[3.2 Other software 5](#_heading=h.26in1rg)

[**4. Design Documents 5**](#_heading=h.35nkun2)

[4.1 Your Sequence Diagram 5](#_heading=h.44sinio)

[**5. Methodology 6**](#_heading=h.2jxsxqh)

[**6. Conclusion 12**](#_heading=h.z337ya)

[6.1 Benefits 12](#_heading=h.3j2qqm3)

[a. Benefits to users: 12](#_heading=h.1y810tw)

[b. Benefits to me: 12](#_heading=h.4i7ojhp)

[6.2 Ethics 12](#_heading=h.2xcytpi)

[6.3 Future Works 13](#_heading=h.1ci93xb)

[**7. References 13**](#_heading=h.3whwml4)

# 1. Introduction

## 1.1 Problem definition

Creating class timetables is an ongoing and labour-intensive challenge faced by academic institutions, typically conducted on an annual or biannual basis (Datta et al., 2005). Historically, this task heavily relied on manual scheduling performed by individuals or teams, consuming substantial time and effort. The intricacy and error-proneness of timetable design have long been acknowledged.

However, this project seeks to address this challenge by introducing automation and minimizing the need for manual intervention. The objective is to streamline the process of timetable generation while reducing the potential for errors. To achieve this, the project will utilize an app that handles the scheduling of courses in a systematic manner.

The app will take each course and its associated preferences into account, systematically inserting them into the timetable. It will carefully consider potential conflicts and overlapping classes to ensure the efficient allocation of resources. Additionally, the app will prioritize the preferences provided, allowing users to indicate their desired time slots or any specific constraints.

By automating the process and incorporating these intelligent features, the project aims to create a more optimized and user-friendly experience for timetable creation. Ultimately, this will reduce the burden on human schedulers, save time, and enhance the overall efficiency of the scheduling process in academic institutions.

## Goals

* Make sure there are no overlapping classes for an instructor.
* No courses in a course schedule are overlapped.
* Make sure a classroom does not have overlapping courses at the same time.
* Make sure the class is big enough for the number of students taking that class
* Make the insertion and deletion to and from the database done by only the admin.
* Ensuring that when an insertion or a deletion to/from the database is done, another insertion/deletion will wait.
* Allowing users (admin and instructor) to see the timetables created.
* Able to handles not just the normal flow of a department course but also in cases of carryover courses.
* Able to compare the timetables of selected students and students in a course.

# 2. Literature Survey

* (Mantri et al, 2015) created a similar project to mine using Java (for the backend) and XML (for the front end). In their application, there will be a need to heavily edit the code to better suit the requirements of different universities whilst with the design I have made for my project, any university will only need. (Mantri et al, 2015) created a similar project to mine using Java (for the backend) and XML (for the front end). In their application, there will be a need to heavily edit the code to better suit the requirements of different universities. With the design I have made for my project, any university will only need to input contents into the database, following the schema already provided.
* (Shashwat Thakur, 2019) created his application for only a faculty making it unreliable for a full university with multiple faculties, as a classroom cannot be used for two different courses in the same faculty at the same time but can be used by two courses in different faculties at the same time as they are treated as separate timetables. to input contents into the database, following the schema already provided.
* (Shashwat Thakur, 2019) created his application for only a faculty making it unreliable for a full university with multiple faculties, as a classroom cannot be used for two different courses in the same faculty at the same time but can be used by two courses in different faculties at the same time as they are treated as separate timetables.

# 3. Background Information

JavaScript will serve as the primary programming language for both the front-end and back-end development in this project. CSS will be used alongside JavaScript for styling and visual presentation. Additionally, a foundational understanding of HTML is required for front-end development.

## 3.1 Required & Used software

* React: front-end framework.
* NodeJs: used for the back end.
* Expressjs: the back-end framework.
* MongoDB: this is the database that will be used in this project.
* Node Package Manager (npm): used for installing and managing the packages and modules used while developing the project.

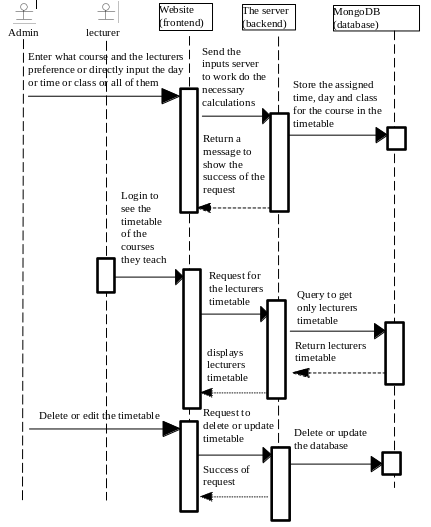
## 3.2 Other software

* Git: the repository for storing my progress.
* Visual Studio Code: code editor of choice.

# Design Documents

## Your Sequence Diagram

This describes the interaction between the users of the system (admins and lecturers) and the system. It also shows the interaction between the multiple modules of the system (frontend, backend, and server which is Mongodb).

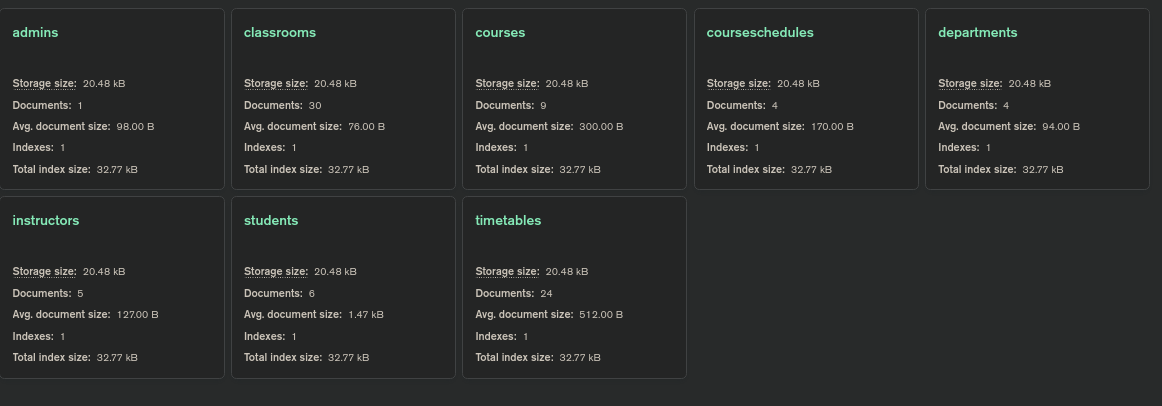


# Methodology

The process of creating class schedules in educational institutions is a challenging and time-consuming endeavour that has conventionally depended on manual scheduling methods. This intricate task involves considering numerous factors such as course requirements, faculty availability, and room availability. However, the reliance on manual scheduling has posed significant demands and required extensive effort from administrators to generate accurate and efficient class schedules.

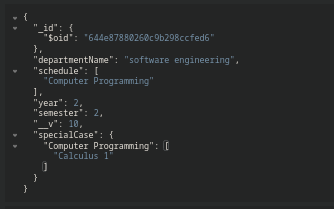
This Project is committed to addressing the aforementioned challenge by embracing automation and reducing the reliance on manual intervention. The primary objective is to streamline the entire process of creating class timetables, making it more efficient and error-free. By automating various tasks and incorporating intelligent algorithms, the Project aims to minimise the need for human involvement and reduce the occurrence of errors. The ultimate goal is to optimize the timetable generation process, allowing for the creation of accurate and well-structured schedules while minimizing mistakes that can arise from manual scheduling efforts.

The initial step in this project involves the creation of a database. The database design employed here is a condensed version derived from a comprehensive university database. Specifically, I selected the crucial components that are highly relevant to developing a functional timetable for the university. I had different schemas for the classroom, instructor, student, department, course schedule (for the departments), course, admin and the timetable itself.



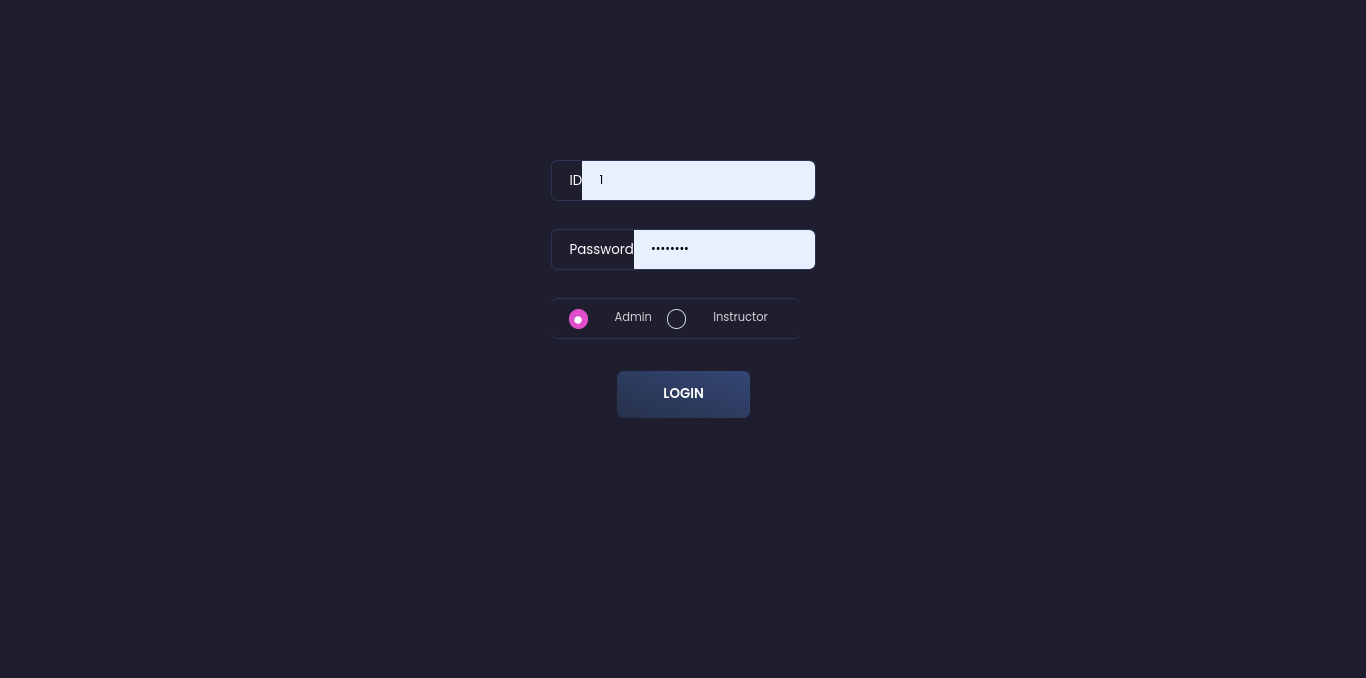
*List of all the tables in the database*

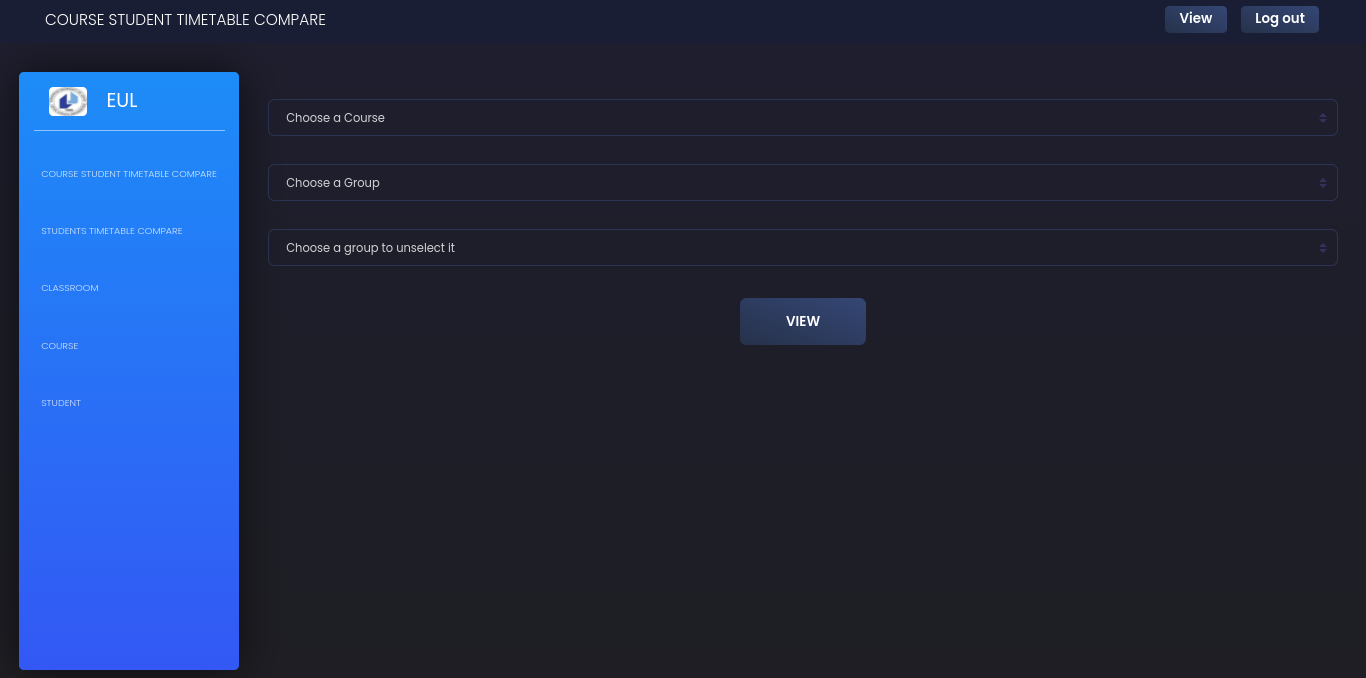
The course schedule is for a department, year and semester. That is, they are normal courses the students of that department and in that year and semester are meant to take. But it has a special-course object that deals with carry-over courses. this works like having a course that will be the main course of interest which a person or people that take that course have a carry-over course and the carry-over course(s) will be inputted as a special course to the initial course and will be checked, during timetable generation for the original course, for any possible overlapping of timetable.



*An example of a course schedule with a special course represented as specialCase.*

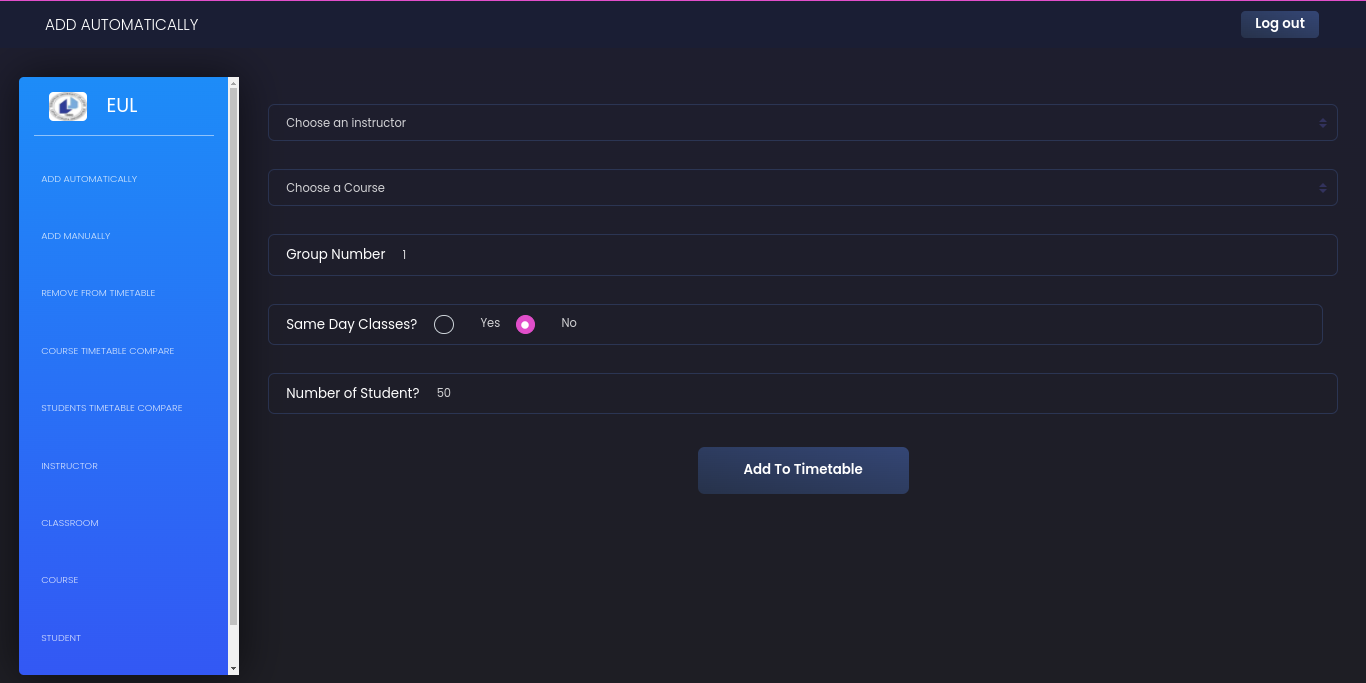
In this project, the instructor has the capability to compare students' timetables within a specific course as well as compare the timetables of selected students. The instructor can also access their own timetable and view classroom assignments, course details, and student information alongside the respective timetables.





*Login page and the instructor dashboard*

On the other hand, the administrator possesses all the functionalities available to the instructor, with the added privileges to manage student records, course listings, instructor assignments, classroom allocations, course schedules, and overall timetables. Additionally, the administrator can determine which courses are assigned to individual students, control the eligibility of departments to enrol in specific courses and manage the inclusion or exclusion of special courses from the course schedule. Lastly, the administrator has the ability to view the timetables of all the instructors in the system.



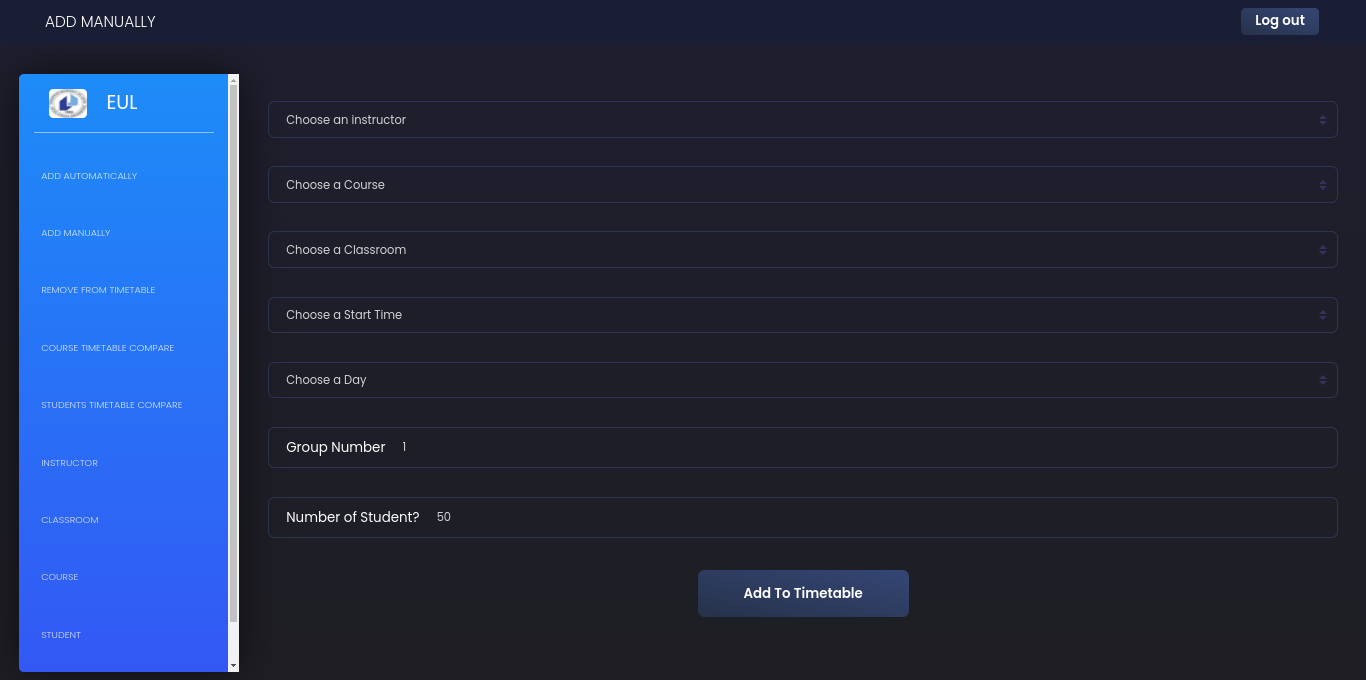
*The admin page to add to the timetable*

There are several algorithms that can be used for automated timetable generation like Constraint Satisfaction Problem (CSP) Solvers (Puttaswamy et al., 2018, 2), Genetic Algorithm (Mittal et al., 2015, 2), Tabu Search Algorithm, et al. But the technique used in this project is a modified Tabu Search Algorithm.

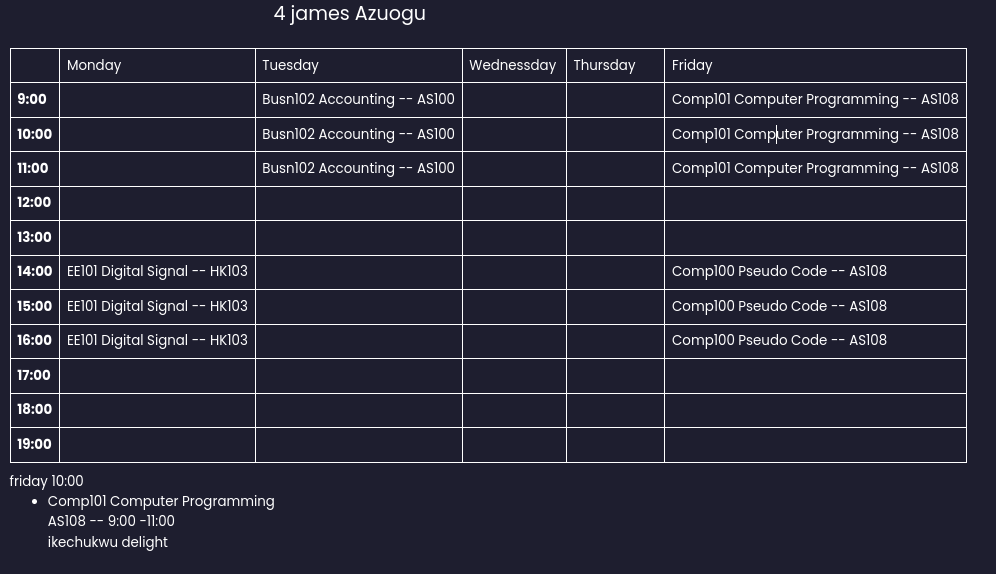
Tabu search is an algorithm designed to find an appropriate time slot for a given course by systematically exploring all available time slots. It starts by examining the first available time slot and checks if the course can fit into it. If it does, that time slot is chosen as the placement for the course. However, as the algorithm progresses, it keeps track of the time slots that have already been visited in a list called the tabu list. This is done to prevent revisiting previously explored time slots and ensure that the algorithm explores new possibilities.

In the modified version used for this project, additional conditions are incorporated to guide the selection of time slots for the course in the timetable. These conditions aim to avoid conflicts and overlapping schedules for both the instructor and the students enrolled in the courses. For instance, the algorithm checks if there are any overlapping courses in the timetable that the instructor or students are already taking, and rejects time slots that would create conflicts. Furthermore, the conditions also involve examining the course schedules, special-course requirements, and availability of suitable classrooms. The algorithm ensures that the chosen time slots do not lead to overlapping timetables, taking into account all the relevant factors to generate a comprehensive and conflict-free timetable solution." to 100 words.

Aside from the timetable creation process being automatic, you can still add to the timetable manually where the admin will choose what class, time and day the course will be taken place. This approach also takes into consideration overlapping courses and special courses.



*Form to add a timetable manually.*



*The timetable of James Azuogu who has a 4 as his student number*

# Conclusion

Having humans do jobs like creating a timetable is an outdated task when it can be easily automated, which is what this project aimed at doing. and with the help of a modified Tabu search algorithm and a few other algorithms it was achieved.

## Benefits

### Benefits to users:

* Reduce the time it takes to create the course timetable for a school
* It eliminates the manual work of creating the timetable
* It reduces the possibility of instructors having overlapping classes
* It reduces the work needed to compare students' free time.

### Benefits to me:

* It gave me an opportunity to learn to use react better.
* It allowed me to work on a project that would be useful in my job application.

## Ethics

When Instructors are solely responsible for creating schedules, there is a risk of human errors, omissions and favouritism. This can result in unequal distribution of resources. However, by automating this task, these risks can be eliminated, ensuring a timetable free from human errors. The algorithms can allocate classrooms, days, and time slots based on predefined constraints and objective criteria. This reduces the influence of personal preferences, errors and overlapping classes for students and instructors, ultimately leading to a more equitable timetable that the university can use.

**Why did I choose this project?**

Over the years I have thought of how difficult it will be to create the timetables for an entire university and now that I can code an app, I think it will be a good idea to automate the procedure and save the person/people who create these timetables some time and strength.

## Future Works

There is definitely room for improvement in the project, some of the improvements are;

* Integrating AI into the project, i.e. making AI choose a much better day and time for the courses in the timetable.
* Making the timetable of lecturers not spread. making them close together in a healthy matter, i.e. a lecturer can have a course on Monday and Wednesday and not like on Monday and Friday.

# References

Datta, D., Deb, K., & Fonseca, C. M. (2005). *Solving Class Timetabling Problem of IIT Kanpur using Multi-Objective Evolutionary Algorithm*.

Mittal, D., Doshi, H., Sunasra, M., & Nagpure, R. (2015). *Automatic Timetable Generation using Genetic Algorithm*.

Puttaswamy, A., Arshad, H. M., Khan, A., V., C. S., & A., P. (2018). A STUDY ON AUTOMATIC TIMETABLE GENERATOR. *INTERNATIONAL JOURNAL OF SCIENCE AND INNOVATIVE ENGINEERING & TECHNOLOGY*.

Younes Messari (2022). Automatic timetable generator. Retrieved from <http://www.aui.ma/sse-capstone-repository/pdf/spring-2022/Automatic%20Timetable%20Generator.pdf>

Siddhesh P. Mantri, Yogesh G. Nipate, Gaurav S. Pawar, Prajyot P. Rumde. Development of Timetable Generation System [TTGS] for College.2010. Retrieved from <https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnx0dGdzMjAwOXxneDphMzNiNDc4ZWQ3ZGFmNjY>

Shashwat Thakur. Automatic Timetable Generation Mini Project Report. 2019. Retrieved from <https://www.scribd.com/document/408248380/229488730-Automatic-Timetable-Generation-Mini-Project-Report-converted-docx>