**EUROPEAN UNIVERSITY OF LEFKE**

FACULTY OF ENGINEERING

Graduation Project I

**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**

24th December 2022

**Table Of Contents**

[**1. Introduction**](#_heading=h.rsw3wq633lnu) **1**

[1.1. Problem definition](#_heading=h.kam805r6fpf0) 1

[**1.1.1. Example-Problems:**](#_heading=h.qfxfeynxvbzb) **2**

[**1.2. Goals**](#_heading=h.qqytvcsb4cgj) **2**

[**2. Sequence Diagram**](#_heading=h.nznbfb371rrb) **3**

[**3. Literature Survey**](#_heading=h.g4icytyi91yc) **4**

[**4. Background Information**](#_heading=h.gz7dgqly5y5p) **4**

[**4.1. Required Software**](#_heading=h.skhm879az1zs) **4**

[**4.2. Other software**](#_heading=h.l73t1mjyulr8) **4**

[**5. Modules**](#_heading=h.yu4jd9rl2tuz) **5**

[**5.1 Frontend**](#_heading=h.jgdjucfi6yj8) **5**

[**5.2 Backend**](#_heading=h.veff01ca3dgz) **5**

[**6. Risk Analysis**](#_heading=h.wh274w7j0kf7) **5**

[**7. Ethics**](#_heading=h.r1vx9p9xb41g) **6**

[**8. Conclusion**](#_heading=h.dzc62fuhrlh9) **6**

[**8.1. Benefits**](#_heading=h.uxuwju2979qd) **6**

[**8.1.1. Benefits to users**](#_heading=h.2fuzdqcwgdy8) **6**

[8.1.2. Benefits to me](#_heading=h.fv5dom27ul49) 6

[**8.2. Future Works**](#_heading=h.iyoetfy6pds4) **6**

[**9. References**](#_heading=h.mtweu1qq3ti4) **7**

# 1. Introduction

## 1.1. Problem definition

Most universities have multiple faculties and departments which have multiple courses taken by the students of these departments and a lecturer can teach more than one course, whether it is from one department or not. They need to have a timetable and creating this timetable can be difficult and time-consuming for humans to create because they have to take into consideration overlapping classes. This project takes that task away from them and reduces the manual work.

The app will take the courses one by one and a preference then inserts them into the timetable taking into consideration overlapping classes and the preference given (if any).

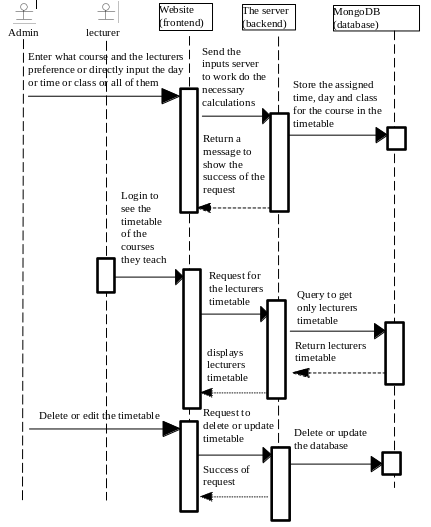
### 1.1.1. Example-Problems:

* There is always a problem with database structure, what fields to have and how to structure it in general, but after a close look at [1], I was able to structure my database in the best way possible.
* Most similar projects tend to just create a timetable with the exception of a means of editing it or inserting a course directly into the timetable.

## 1.2. **Goals**

* Make sure every course has a slot in the timetable.
* Make sure there are no overlapping classes for an instructor.
* 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

# 2. Sequence Diagram



# 3. Literature Survey

* [2] created a similar project to mine using Java (for the backend) and XML (for the frontend). 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 to input contents into the database, following the schema already provided.
* [3] 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.

# 4. Background Information

The programming languages that will be used in this project are JavaScript (for both the frontend and the backend) and CSS. There is also basic HTML knowledge needed for the frontend.

## 4.1. **Required 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.

## 4.2. **Other software**

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

# 5. Modules

## 5.1 **Frontend**

The front end focuses mainly on what the user sees and interacts with but also takes into consideration the connection to the server to access the databases.

* React will be used to create the structure of the site and handle all the connections to the server. React is one of the most popular javascript front-end frameworks because of its simplicity and easy-to-find encountered problems due to its large community support.
* CSS will be used for styling the app, making it look good and presentable for the users.

## 5.2 **Backend**

The backend is the server that handles the databases and where all the algorithms for scheduling the timetable will be stored. The front end can send requests to already created routes in-order to insert or retrieve to/from the timetable in the database.

* Expressjs is a prevalent Javascript back-end framework that is easy to understand and has very big community support, hence it will be a very good framework to use for the back-end of this project
* MongoDB is a NoSQL database that can be easily handled with a mongoose package/module. It is the database I have chosen to work with in this project.

# 6. Risk Analysis

* Having overlapping courses taught by the same lecturer
* Inputting two or more courses in the same class at the same time

# 7. Ethics

When employees are in charge of creating the schedules, there is a strong risk of encouraging clientelism and favouritism. Using software to accomplish this activity can eliminate these risks and guarantee that the provided timetable is unbiased.

# 8. Conclusion

## 8.1. **Benefits**

### 8.1.1. 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

### 8.1.2. Benefits to me

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

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.

## 8.2. **Future Works**

I could continue to work on this project to make it bigger and possibly integrate students into the project, i.e. I will not only be checking for overlapping instructors’ classes but also for students as well.

I could also make a way for instructors to compare multiple students' timetables to find their free time to insert labs, quizzes, or other activities.

# 9. References

[1] Younes Messari (2022). Automatic timetable generator. Retrieved from [Automatic Timetable Generator Capstone Design Spring, 2022](http://www.aui.ma/sse-capstone-repository/pdf/spring-2022/Automatic%20Timetable%20Generator.pdf)

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

[3] Shashwat Thakur. Automatic Timetable Generation Mini Project Report. retrieved from [Automatic Timetable Generation Mini Project Report | PDF | Feasibility Study](https://www.scribd.com/document/408248380/229488730-Automatic-Timetable-Generation-Mini-Project-Report-converted-docx)