

**Course Offering**

by

**Mohammad Salhab**

**Abed Al Rida Nehme**

**Wafik Ibrahim**

**Ibrahim Dahboul**

**Ali Abou Khashfe**

**Mohammad Ghoush**

**Project**

Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor

in Computer Engineering

Department of Electrical and Computer Engineering

Faculty of Engineering

3rd Year

Contents

s[List of Figures 4](#_Toc194665529)

[List of Tables 5](#_Toc194665530)

[Project Description 6](#_Toc194665531)

[Project Overview 6](#_Toc194665532)

[Objective: 6](#_Toc194665533)

[Background: 6](#_Toc194665534)

[Literature Review 8](#_Toc194665535)

[Top Level View 8](#_Toc194665536)

[Historical review 8](#_Toc194665537)

[Optimization Techniques Used 9](#_Toc194665538)

[Research Gaps and Opportunities 9](#_Toc194665539)

[Conclusion 10](#_Toc194665540)

[Applications 10](#_Toc194665541)

[Alternative Designs 10](#_Toc194665542)

[Project Planning 11](#_Toc194665543)

[Constraints 11](#_Toc194665544)

[Implementation Environment of the Current System: 11](#_Toc194665545)

[Partner or Collaborative Applications: 11](#_Toc194665546)

[Off-the-shelf Software: 12](#_Toc194665547)

[Anticipated Workplace Environment: 12](#_Toc194665548)

[Schedule Constraints: 13](#_Toc194665549)

[Budget Constraints: 13](#_Toc194665550)

[*Project Issues* 13](#_Toc194665551)

[Team Members Tasks 14](#_Toc194665552)

[Ethical Issues 14](#_Toc194665553)

[Software Model Process: 14](#_Toc194665554)

[Feasibility Study 15](#_Toc194665555)

[Tools/Technology 15](#_Toc194665556)

[Milestones 15](#_Toc194665557)

[Requirements 17](#_Toc194665558)

[Use Cases 17](#_Toc194665559)

[Functional Requirements 17](#_Toc194665560)

[Data Requirements 17](#_Toc194665561)

[Non-Functional Requirements 18](#_Toc194665562)

[Design 20](#_Toc194665563)

[Class Diagrams 20](#_Toc194665564)

[Dynamic Model 20](#_Toc194665565)

[Subsystem Decomposition 20](#_Toc194665566)

[Hardware / software mapping 20](#_Toc194665567)

[User Interface 20](#_Toc194665568)

[Test Plans 21](#_Toc194665569)

[Implementation 22](#_Toc194665570)

[Results Evaluation 23](#_Toc194665571)

[Conclusion 24](#_Toc194665572)

[Summary 24](#_Toc194665573)

[Novelty 24](#_Toc194665574)

[Integrity and Values 24](#_Toc194665575)

[Future Work 24](#_Toc194665576)

[References / Bibliography 25](#_Toc194665577)

[Appendix 27](#_Toc194665578)

# List of Figures

# List of Tables

# Project Description

## Project Overview

The project focuses on a local hosted platform that generates the Course Registration Numbers (CRNs) and timetables for academic institutions, such as universities and schools. The process is done by applying Genetic Algorithm (GA) on the course’s data that is provided by the tutors through a restricted yet smooth and user-friendly web interface that is supported by a database holding all the necessary info. Ensuring sufficient allocation of resources such as classrooms, Dr’s name and time slot.

## Objective:

1. Automate the process of CRNs generation and timetables using Genetic Algorithm.
2. Reduce exponentially the manual effort and time spent by tutors in creating timetables, where all it is expected from them is to provide the needed data.
3. Optimize resource allocation and minimize schedule conflicts.
4. Provide a flexible and more customizable system for academic institutions making it more adaptable for both tutors’ and students’ specific needs and exceptional cases.
5. Ensuring the scalability of the system on various institutions with different sizes and complexities.
6. User-friendly web interface where the usability features are concise and clear for all potential users regardless of their background.

## Background:

University time tabling is a process that undergoes a lot of constraints and limitations and needs optimizing resources efficiently. The growing demand for more flexible, user-friendly, and conflict-free scheduling systems has made this a subject of ongoing research. The process itself is not only limited to logistical problems and faculty availability but also related to the needs of the students and the instructors. All these factors make the university course time-tabling a challenging optimization problem.

The field of academic administration, specifically **timetable scheduling**, is a time-consuming and complex process that requires coordination between the academic administrators to allocate classes and labs for multiple courses. This process involves multiple variables and considerations such as classroom availability and capacity, course/lecture type, faculty availability and student preferences. Therefore, manually creating timetables is not just labor-intensive and complex but also prone to conflicts.

The problem is clear, but since we needed a starting point, we had to ask these questions:

* “How can we engineer a time tabling system that balances the needs of the instructor and the student, ensuring an efficient and ***conflict-free schedule***?”
* “What are the trends int the time tabling field and what options does universities rely on when it comes to***modern course offering system***?”
* “How can we optimize the time tabling to improve efficiency and accuracy of university *course* ***timetabling systems***?”

These questions serve as the foundation for investigating various optimization methods, such as meta-heuristics and machine learning techniques, and their impact on the real-world solution for the scheduling problem. So, we had to set boundaries for the studies that will be shaped by these factors:

* Time frame: we wanted to spend about **3-4 hours** revising the research since we didn’t have much time and we were already similar with the project idea and the choices we wanted to make, and since optimization is an old trend so we included the studies from the last **10-15 years.**
* Geographical Scope: We decided not to limit our focus on a certain region, and we made it **global**.
* Types of Studies: we mainly focused on the studies that included time tabling and the optimization techniques to make time tabling as efficient as possible. So, we focused on case studies of **existing timetable systems**. And we mostly included research about real life applications.

To handle such issues, an advanced algorithm like **Genetic Algorithm (GAs)** can be a powerful tool to automate and optimize such process promising to provide the global optimal timetable tailored for students’ and tutors’ preferences.

**Genetic Algorithms** are an evolutionary algorithm inspired by the natural selection and genetics principles in the real world of constant evolution. They work by continuously evolving a collection of potential solutions to a problem, using operations/functions like selection, crossover, and mutation. GAs explore numerous solutions to find optimal ones while considering the multiple given constraints which makes it well-suited for such a problem. Their ability to handle such non-linear, dynamic and multi-constraint problem makes them ideal for generating tailored conflict-free timetables.

In the case of **course offering**, GAs will optimize the allocation of resources (e.g. classrooms, labs, time slots, tutors’ availability) all while considering the constraints like course prerequisites, tutor availability, and student preferences. By implementing such algorithm, academic institutions could generate effortlessly and automatically timetables significantly reducing the manual effort, enhancing the experience and provide a highly adaptable and scalable solution, allowing institutions to customize this algorithm to their specific need and constraints.

## Literature Review

### Top Level View

Universities usually set their timetables based on various constraints and limitations, these limitations can vary between departments. For example, a department may have many instructors and professors and have a very nested study plan that will of course increase the complexity of the process.

The core challenge in timetable lies in balancing the needs and preferences of **students**, **instructors**, and **universities,** with taking into consideration that each one of them might have its own priority. Instructors may have limited availability, while students often have specific course preferences or constraints based on their individual schedules.

We noticed that the timetable problem can be highlighted by the following key constraints:

* Instructor availability.
* Room preferences.
* Room allocation.
* Course pre-requisites.
* Time-slot restrictions.

### Historical review

University timetable has evolved from **manual scheduling** and basic **heuristics** to more advanced **optimization techniques.** Now we mostly use genetic algorithms and the meta-heuristics for handling complex constraints efficiently.

**Current Trends:**

AI and **machine learning** are increasingly used for real-time, dynamic scheduling, while **constraint programming** and **meta-heuristics** help generate conflict-free timetables. Also, we noticed that most designs are focused on the user interface and making it easier and user friendly.

**Relevance to Our Project:**

Our project aims to create an optimized, conflict-free timetable system by using advanced algorithms to meet the needs of both students and instructors while adapting to real-time data.

**Key Themes of Literature**

Optimization Algorithms for University Timetabling

* Overview: Common optimization techniques include **genetic algorithms**, **simulated annealing**, and **meta-heuristics**, which help generate conflict-free timetables and efficiently allocate resources.
* Relevance: These techniques can optimize your course offering system by resolving conflicts, ensuring efficient room allocation, and handling multiple constraints.
* **Studies to Reference**:
  + *"A Review of Optimization Algorithms for University Timetable Scheduling"*
  + *"Optimization Techniques in University Timetabling Problem"*
  + *"Meta-heuristic approaches for the University Course Timetabling Problem"*

Challenges in University Timetabling

* **Common Challenges**: we have constraints concerning the student, the Doctor, the lab instructor, university requirements, study plan.
* **Relevance**: Our system can address these by using optimization algorithms to resolve conflicts and ensure efficient use of resources.
* **Studies to Reference**:
  + *"School timetabling for quality student and teacher schedules"*
  + *"Optimization Techniques in University Timetabling Problem"*

User-Centered Design in Timetabling Systems

* **Balancing Needs**: Effective systems balance the needs of both students and instructors by considering their availability and preferences.
* **Relevance**: Your system can incorporate user input to allow customization, improving user satisfaction (like letting the dr decide what hours he prefers)
* **Studies to Reference**:
  + *"School timetabling for quality student and teacher schedules"*
  + *"A Survey of University Course Timetabling Problem"*

### Optimization Techniques Used

1. **Heuristic Methods**:
   * *Simple, rule-based approaches like* ***greedy algorithms*** *are fast but may not guarantee optimal solutions.*
2. **Meta-heuristics**:
   * *Techniques such as* ***genetic algorithms****,* ***simulated annealing****, and* ***tabu search*** *are more effective for large-scale problems, generating high-quality solutions by navigating complex constraint spaces.*
3. **Constraint Programming**:
   * *Models the timetabling problem with a focus on constraints (e.g., room, time, availability) and utilizes search techniques for precise solutions.(we previously did this in the basic version we created as a project for the programing two course)*
4. **AI and Machine Learning**:
   * *AI methods like* ***reinforcement learning*** *and* ***machine learning*** *help optimize timetables dynamically by learning from historical data and adapting in real-time.*

### Research Gaps and Opportunities

* ***Limited Real-Time Adaptation****: Most systems still struggle with real-time scheduling adjustments. There’s an opportunity to develop* ***AI-driven systems*** *that adapt dynamically to changes in instructor or student availability. (we can include the information we learnt from the Introduction to Ai to make an Ai module that is able to learn how to construct the time table like a human) (****he can’t replace the efforts of both Dr. Imane and Dr. Hiba****)*
* ***Integration of User Preferences****: While user preferences are often considered, integrating* ***real-time feedback*** *from both students and instructors into the scheduling process remains underexplored. (for example, we have courses that is always set by other departments for general engineering requirements)*
* ***Scalability and Efficiency****: Many current systems face scalability issues when applied to larger universities. There’s a gap in creating* ***highly scalable*** *optimization methods that can handle large datasets efficiently.*
* ***Hybrid Approaches****: Combining multiple optimization techniques, such as* ***AI*** *with* ***meta-heuristics****, could offer better solutions, but more research is needed on* ***hybrid models****.*

### Conclusion

University course timetabling remains a complex problem that involves balancing multiple constraints such as instructor availability, room allocation, and student preferences. Optimization techniques like **meta-heuristics**, **genetic programming**, and **AI** are crucial in developing efficient timetables. Despite multiple researches and papers, challenges such as real-time adaptation and scalability persist. However, these challenges also present opportunities for future research, particularly in areas like **AI-driven scheduling** and **hybrid optimization models**.

This review highlights the importance of integrating advanced optimization methods to improve timetabling systems, ultimately making them more flexible, scalable, and adaptive to real-world needs

## Applications

This GAs-based project is highly reusable and can be used in various scenarios. In educational institutions like BAU, it streamlines and automates the production of university timetables and CRNs with maximum resource usage by efficient allocation. It eliminates significant manual effort and can accommodate changes easily, for instance, adding new courses, making room for irregular students, or adjusting tutors' availability.

Other than education, the system can be applied in a variety of other fields. For example, it can schedule employee training sessions, hospital staff shifts, or conference events. In short, it's a all-purpose solution for any complex, non-linear problem with many variables and intricate constraints.

## Alternative Designs

1. Ant Colony Optimization (ACO) – ACO mimics the ant behavior while searching for food. where each ant will explore paths to find the solution. Over time the ants will find the shortest paths reinforcing good solutions while weaker paths will fade over time
2. Cuckoo Search Algorithm (CSA) inspired by cuckoo birds, where each solution will represent an egg in the nest the new egg (solution) will replace the worst ones, ensuring continuous improvement.
3. Swarm Intelligence (SI) each solution will represent a particle. A group of solutions moves around a search space each solution adjusts its position based on its best-found solution
4. Tabu Search mimics human problem solving by remembering past choices and avoiding repeating bad one.

# Project Planning

## Constraints

### ***Implementation Environment of the Current System:***

The system is expected to be implemented in the computer engineering department and will be utilized personally by the faculty and administrative staff. Due to its lightweight nature, the system is able to operate smoothly without powerful servers or specialized hardware. The frontend consumes HTML, CSS, JavaScript, and PHP and can be opened by any normal web browser while the development is done in Visual Studio Code (VS Code). The backend is MySQL running on Google Cloud which provides dependable and scalable data management. This puts the system readily to ease into a department’s infrastructure, while providing a simple and friendly interface for effective timetable management.

### Partner or Collaborative Applications:

Building a MySQL database, integrating it with a webpage, and providing multi tech access and data manipulation showcases a collaborative effort towards building an effective and dynamic system for timetable management and generation. This combines various web development technologies, namely: HTML, CSS, JavaScript, PHP, and additionally uses Python for data manipulation and interaction. Each technology is essential to the proper functioning of the system which in turn enables the collaboration between the user interface, data base management system, and backend business logic.

1. ***MySQL Database Management:***

MySQL has a database that acts a backbone for this system as it enables the storage and management of all timetable data. The database takes care of record maintenance with respect to course titles, teachers, timetable information, and other relevant factors of the timetable. However, MySQL, as a highly efficient Relational Database Management System (RDBMS), allows the user to execute queries, and retrieve or modify data without any hassle.

In this case, the schema of the database is constructed to accommodate all possible information relating to courses, personnel, and the timings of classes. For instance, a particular table can have fields with course titles, instructor titles, days of the week, and the starting and finishing times for each class. This database eliminates the headache caused by adding, editing, or deleting courses and guarantees that data is stored systematically.

1. ***Web Interface with HTML, CSS, JavaScript, and PHP:***

A graphical user interface is designed with a blend of HTML, CSS, JavaScript, and PHP, all of which work in conjunction with a MySQL database. These technologies allow for the effortless insertion of course data into the database and extraction of timetable queries.

• With HTML, structure of the web page is created to enable users to provide important information such as the course, instructor and the respective timings. The provided data is captured by an HTML Form which is then forwarded to a backend PHP script for processing.

• The usage of CSS is to style the interface of the webpage so as to improve the overall user experience. The webpage is modified in a way that makes it attractive and easy for the users to navigate.

• JavaScript enables the clients to submit data after it has been validated and also enables additional interactivity. Javascript allows for the submission of data dynamically, and facilitates updates to be displayed without refreshing the web page.

• PHP, which is the server side scripting language, takes care of the logic behind form submissions; validates the forms and then interacts with MySQL to either insert or update timetable data. PHP scripts interact with the database and send the results to the frontend which can then be used by the system to display the current timetable data.

1. ***Integration with Python for Data Processing:***

Python is essential in accessing the MySQL database, used for more complicated automated tasks, such as handling the timetable, and manipulating the data into a format that meets the needs of the end user, all being outside the capabilities outlined in the web interface. This system can query the data directly from Database using Python’s MySQL connector and create reports, visualization, and automated schedules.

Due to its versatility, Python can also handle more complex manipulations of data if needed when generating the timetable, such as formatting of data into a user-friendly timetable, overlap calculations, or reporting in different formats (PDF, CSV, …). In addition, the Python can also be used to automate processes such as sending notifications or reminders using the

1. ***Collaborative Nature of the System:***

The system combines MySQL, web technologies, and Python to make the parts work well together. MySQL acts as the core, keeping all course information safe and easy to find. The web interface lets users easily access and interact with this information. Python allows for analysis and processing of the data, making the whole system stronger.

This setup keeps data consistent and easy to reach on different platforms. Every part, whether it's the database, the web interface, or the Python backend, works together to ensure a smooth user experience. Also, the system can grow and adapt over time. It can add more features, like advanced scheduling, interactive dashboards, and the ability to connect with other systems used in schools or institutions.

### Off-the-shelf Software:

Ready-made software solutions make developing software easier by providing reliable tools that have already been tested and proven to work well. For this particular project, we're using MySQL, Google Cloud, and Visual Studio Code (VS Code) to handle important tasks like managing the database, hosting the application, and creating a good environment for development.

- \*\*MySQL\*\* is a free system that helps organize databases. It’s great for storing things like timetable data because it makes finding and updating information fast and keeps everything correct and efficient.

- \*\*Google Cloud\*\* offers flexible online services. One of these is Cloud SQL, which can host the MySQL database. Google Cloud provides a safe and cost-effective way to host applications, and it offers tools to expand the application as needed, ensuring it’s always available and dependable.

- \*\*Visual Studio Code (VS Code)\*\* is a simple yet powerful code editor. It works with many programming languages and helps developers be more productive with features like auto-completion of code, tools for finding and fixing errors, and version control to keep track of changes in the code, making it an ideal choice for web development.

Together, these off-the-shelf solutions help make the development process smoother, boost efficiency, and ensure that the system can grow and remain secure over time.

### Anticipated Workplace Environment:

This project is for the Computer Engineering department, specifically to aid one of its faculty members or doctors. The application is crafted to be simple and efficient, capable of running smoothly on a personal computer without needing expensive hardware or lots of resources. It can operate daily on standard desktop or laptop computers commonly found in academic settings.

The application will feature a straightforward design, making it user-friendly. This ensures that the doctor can use it without needing much training or technical expertise. The main goal is to simplify tasks like managing schedules, entering course details, and creating reports with ease. The easy interface helps shorten the learning time, allowing the doctor to focus on their primary responsibilities.

The system is lightweight, so it will be quick even on computers that aren't the newest or fastest. This is particularly important in schools where time is often limited, and efficiency is key. The system will blend seamlessly into the department's daily activities, offering a practical and reliable solution for managing course offerings without requiring much technical help.

In summary, the project is designed for a typical academic environment, where being easy to use, fast, and accessible is essential. The lightweight design and simple interface will ensure it effectively supports the doctor’s administrative tasks, making their work smoother and more efficient.

### Schedule Constraints:

The schedule and deadlines for this project are set up to be flexible, so we can adjust things as needed while we develop the project. Because the project is not too big and has clear goals, we do not expect to face major challenges that could slow us down. This flexibility means we can handle unexpected problems or changes without putting too much pressure on the team. We also have enough time to test everything, fix any problems, and make improvements to ensure the final product is up to standard. This way, our deadlines are realistic and allow us to keep moving forward steadily while making sure the project stays high-quality and works well.

### Budget Constraints:

This project has a rule: we must use only open-source software. This type of software offers many advantages like saving money and being able to change tools to fit our needs. For example, we use MySQL to manage databases, PHP for backend work, and Visual Studio Code for coding. These tools are free to use, and because many developers around the world use them, we can get support and updates from this community.

Open-source software comes with certain responsibilities too. We need to work within what these tools can do and make sure everything works well together. We avoid software that requires a paid license. By doing this, we can be more creative and have more control over how we build things. However, we might face some technical challenges from time to time, as we are not relying on commercial software solutions. These challenges push us to find solutions within the open-source community, which can lead to innovative outcomes.

## *Project Issues*

*Issues that have been raised and do not yet have a conclusion.*

Avoiding Conflicts:

The project's goal is to steer clear of any problems with classes scheduling and resource allocation in order for things to run well it is imperative that room assignment professors and classes do not overlap our system facilitate the prompt and identification of scheduling conflicts the system alert alerts us right away if there is a problem adding new classes or events so we can fix it before it becomes a problem teachers and administrators canmore easily modify schedules as needed thanks to this proactive approach the department runs more smoothly once this effective system is in place since the possibility of errors is reduced.

Furthermore, schedules may be easily updated and modified thanks to the technology which reduced the likelihood of inadvertently creating problem that if this effective strategy ensures that the department runs smoothly.

Web-Database Connections:

the web page and database must have a solid connection for the project to succeed. This makes it casier for us to obtain, enter and update data. HTML, CSS, JavaScript and PHP will be used in the development of the web page all of the course data will be managed and stored by MySQL database, which of this page will communicate with. PHP will operate as an essential connector guaranteeing that every user activity on the page may efficiently interact with and update the database. This configuration ensures that the database and web page function together, making managing course information simple.

We will connect to the database via secure techniques because maintaining accurate data and an effective system are essential. This connection will remain stable even if numerous users are accessing or updating the system simultaneously this configuration ensures that any modifications made to the web page are immediately reflected in the database providing everyone with current and correct information.

Multi-Lecturer Courses:

Courses with several lecturers can be handled by the system. This implies that the timetable must account for the possibility that various lecturers will teach various sections of the same course. Administrators can designate who teaches each section or time slot and assign multiple lecturers to a single course. This helps prevent scheduling problems and ensures every lecturer is included. The timetable also lets you see and manage all instructors' schedules together, preventing any conflicts in their teaching times. This makes it easier to handle courses with multiple instructors.

General Engineering Courses:

The system is designed to handle courses with more than one teacher. When a course has multiple instructors, it can assign different teachers to different sections or times. Administrators can select different teachers for each part of a course and assign specific sections to each teacher. This method keeps classes structured and helps avoid scheduling conflicts, particularly when several teachers are participating. The timetable is also practical and effective since it gives administrators a clear picture of the ability to manage all of the teachers' schedules in one location. This keeps their teaching responsibilities from overlapping.

## Team Members Tasks

Web-development: Ali Abou Khashfe and Abed el Rida Nehme

Algorithm: Mohammad Salhab and Ibrahim dahboul

Database and connections: Mohammad Ghoush and Wafik Ibrahim

## Ethical Issues

Confidentiality in this project, we received important documents that help us manage schedules better. These documents have detailed information, such as class schedules, assigned teachers, and the institution's rules. They are kept confidential because they ensure the system meets the needs of the department.

We make sure this information stays safe. Only authorized people can see these documents, and we use them only to improve how the timetable system works. We protect both digital and paper copies, making sure they are not shared with anyone outside the project team.

By doing all this, we follow privacy rules and keep the department’s trust in us.

## Software Model Process:

For this project, we decided on the Waterfall model for software development because our needs were clear right from the start. We believed our needs would stay the same throughout the project. We decided to use the Waterfall model because it allows us to work through the project steps one at a time. This way, we can follow the right sequence: gathering requirements, designing, developing, testing, and finally deployment. We started by meeting with Dr. Iman Haidar and Dr. Hiba Bazzi. These discussions were crucial to understanding the complete project. We talked in detail about the timetable management system, covering all the important features we needed, the goals we wanted to achieve, and any limitations we had to pay attention to. After agreeing on these requirements, we proceeded to the design phase. During the design phase, we focused on making an Entity-Relationship (ER) diagram for our MySQL database. This diagram was essential because it helped us organize and structure the data for the timetable system effectively.

It showed how different elements like courses, instructors, and schedules were interconnected. This guided us in setting up the database. At the same time, we began planning the website layout and functionality to ensure it was user-friendly.

We also pinpointed which Python libraries we needed for backend processing, data manipulation, and database integration. Our team took the time to learn and understand the libraries so that we could use them well when implementing the system. We used the Waterfall model to guide us, which allowed us to set clear goals for each step Before beginning the following stage, we made care to finish the previous one completely. We were able to create each component of the system in a methodical and structured manner thanks to this technique. After obtaining all required information, we proceeded to design and finally executed the final implementation.

## Feasibility Study

Making a schedule for the entire university is a significant task that requires a lot of time and work. We choose to concentrate just on the Computer Engineering Department in order to simplify this process. This decision guarantees that we can finish the project in the allotted time and helps us manage it more effectively. By concentrating on just one department, we can simplify the process and develop a system that truly meets their needs. This approach gives the department a solution that is both practical and efficient. Focusing more narrowly allows us to pay attention to details and ensure we meet all the special scheduling needs specific to the Computer Engineering Department.

On the timeframe for our project, we intend to complete everything prior to the commencement of the final exams. We want the timetable system to be operational before the start of the next academic term, therefore this deadline is crucial. Teachers and students would be able to efficiently plan their schedules if the technology is available by then. To achieve this deadline, we have divided the project into phases. Each phase gives us enough time to finish important tasks like gathering requirements, designing the system, developing it, testing it, and finally launching it. By following this plan, we hope to successfully finish the project in the given timeframe, providing a working and useful timetable system to the Computer Engineering Department before the academic year ends.

## Tools/Technology

In order to keep the database operating effectively, accommodate additional users as necessary, and preserve security and availability, the project makes use of MySQL hosted on Google Cloud. In order to construct the project's user interface, HTML, CSS, and JavaScript are used. These technological advancements guarantee that the website is visually appealing and user-friendly. On the backend, where users don't see but where important operations happen, PHP is used. PHP is responsible for managing the communication with the server and handling interactions with the database. All development is carried out in Visual Studio Code (VS Code), a lightweight code editor that supports the required technologies, offering features like debugging, syntax highlighting, and version control integration to streamline the development process.The python code on pycharm.

## Milestones

The process to build the course offering and timetable management system is organized into clear steps. Each of these steps is important to make sure the project is finished successfully. Here are the main steps that help the project move forward:

*1. Requirement Gathering and Analysis:*

The first big task is to have a meeting with key people, such as faculty members and administrators. The aim of this meeting is to gather detailed information needed for the project. We need to clearly understand the main features required for the timetable system. These features include scheduling courses, assigning lecturers, detecting any conflicts, and creating the timetables. All of these requirements are documented thoroughly. This ensures everyone involved has a clear understanding of what the project involves and what the goals are.

*2. System Design:*

Once the requirements are clear, the design phase kicks off. During this phase, the team creates an Entity-Relationship (ER) Diagram for the MySQL database, which helps to organize all course-related data clearly. The overall system design is set up, ensuring that the frontend, backend, and database will work smoothly together. Wireframes or models for the user interface (UI) are developed to show what the website will look like. The team chooses specific programming languages and tools, such as HTML, CSS, JavaScript, PHP, and MySQL, to build the system.

*3. Database Setup and Configuration:*

In this step, the MySQL database is installed on Google Cloud. Its structure is based on the ER Diagram, and tables and connections are carefully organized. Security measures, like user login and data protection, are added to keep the database safe. The team conducts tests to ensure the database connects properly and handles queries effectively.

*4. Frontend Development:*

The frontend development starts with building the basic structure of the website using HTML. CSS is employed to style the site, ensuring it looks appealing and works well on various devices and screen sizes. JavaScript is utilized to add interactive elements to the system, such as course searches and alerts for scheduling problems. Thorough testing ensures the frontend is user-friendly and functions smoothly across different browsers and devices.

*5. Backend Development:*

The backend development uses PHP to handle user inputs, interact with the MySQL database, and manage course schedules. CRUD operations (Create, Read, Update, Delete) are put in place to let users manage courses, teachers, and schedules efficiently. A conflict detection algorithm in Python identifies and prevents scheduling overlaps, ensuring that no courses clash in the timetable. Data checks and error handling are implemented to ensure reliable backend operation.

*6. Integration and Testing:*

In this phase, the frontend is connected to the backend, allowing users to interact with the system. They can add or modify courses easily and use the database without issues. Comprehensive testing is carried out to verify that all essential functions work correctly, including scheduling, teacher assignments, and timetable generation. Additional usability and performance tests ensure the system is easy to use and meets performance standards. Any problems discovered during testing are promptly resolved.

*7. Deployment and Final Review:*

Here, the system is launched into a real environment, such as a university server or a cloud service. All components, including the database and web server, are properly configured for production. A final review with stakeholders confirms that all requirements are met, and the system is working as expected. Once approved, it is made available to the Computer Engineering Department for use.

*8. Post-Deployment Support and Maintenance:*

After the system goes live, continuous support and maintenance are provided to keep it operational. The system's performance is closely monitored to ensure efficiency. Any bugs or issues reported by users are addressed swiftly. Based on user feedback, minor improvements or updates may be implemented to enhance the system's capabilities and user experience

# Requirements

## Use Cases

This section begins to describe in more specific and precise detail exactly what steps the system takes in the course of its performance. Use cases serve not only to more specifically define the system (and its boundaries), but also to identify functional requirements, to identify initial objects / classes, and to organize the work.

## Functional Requirements

* **User Authentication:** To enforce security in the system the Tutors (Drs) must log in using valid credentials that are provided to each one of them uniquely. This step verifies validity and authentication of the user where the Dr can then input the course and classroom data needed to create the required timetable.
* **Course Input Interface:** The system should have an easy-to-use interface where the Drs can input some course-related details including course code (COMP364 for example), course title, number of sessions, credit hours, verify if the course have a laboratory or only a lecture-based course, and the course capacity.
* **Classroom Availability Management:** Tutors can input available classrooms and their capacity which will later be handy as the rooms are assigned to courses according to the course capacity.
* **Tutor Input Interface:** The system provides a full page for the admin (and possibly some user/Tutors) to input the university’s Tutors details (full name, their respective major…etc.) and their log-in credentials.
* **Timetable Generation:** The system uses a genetic algorithm to generate conflict-free timetables based on the entered data after applying multiple selection and mutation cycles on a population of schedules.
* **CRN Assignment:** Assigns unique Course Reference Numbers (CRNs) for each course in the timetable based on the input data and explore multiple options and timetables variation which couldn’t be possible if made manually.
* **Timetable Viewing and Exporting:** Tutors can view the end-result, print, and download their generated timetables.
* **Admin Panel:** Admin users can manage accounts (Tutors log-in credentials), view system usage, and input some initial required data to feed the database including available rooms, some course details, tutors accounts credentials…etc.

## Data Requirements

* **Tutor Data:** Names, IDs, department, and login credentials, and the major they are responsible for. This data will be used by the system to authenticate the users and link them to the course they provide and teach.
* **Course Data:** Course codes, titles, number of weekly sessions, prerequisites (if exists), and the respective assigned tutors. This data type is essential for the algorithm to generate a conflict-free timetable.
* **Classroom Data:** Classroom locations (EB141 for example), capacity, available days/hours, a lecture hall or a laboratory room, what kind of lab (if it was), building name. This data will be used later to assign rooms to courses according to their respective capacities.
* **Historical Data:** Past timetables from previous semesters that are manually made could be very useful to improve future scheduling. This data can help identify recurring issues and patterns, like frequent room conflict that happened in the past. Therefore, the system will make smarter decisions and reduce exponentially the computational effort and time. It also improves consistency along academic terms.
* **Study Plan Computer Engineering:** This will improve the expected output but will also still take into consideration some exceptional cases and scenarios for irregular students for example which leverage this system over other traditional and manual scheduling.
* **Availability Time For each Tutor:** The availability time for each Dr is used to assign them to the course that best fit their time making sure there is no overtime work that my overwhelm the Tutors.

## Non-Functional Requirements

Our system must meet certain quality standards. These non-functional requirements ensure the software is reliable, secure, maintainable, and user-friendly.

1. **Performance Requirements:**

Performance is crucial to make sure the system functions efficiently and responsively, even under high load and excessive and numerous data which are expected in our project where our system is subject to essential university credentials.

1. **Dependability Requirements:**

Even though our final output isn’t taken for granted and rechecked by the Drs, it is still essential for our system to be dependable and consistent that performs exactly as intended. These requirements include fault tolerance, independency of the system where it is not dependent on any external APIs that may comprise essential university data.

1. **Maintainability and Supportability Requirements:**

To allow future updates and troubleshooting, the system is designed with maintainability taken into consideration. Our software is modular and broken down into multiple small sized files for this exact purpose, making sure that the future maintability process is quick and smooth.

1. **Security Requirements:**

Each user (tutor) has unique log-in credentials to ensure authentication access only. Also, the system’s data is secured and hashed in a protected Database environment.

Furthermore, the system doesn’t require any sensitive information from the users (only asks for data that is mostly available to the public), thus no harm will be conflicted even if this system is compromised.

Finally, the software’s output is reconsidered and rechecked by the Dr that may decide to use it as a final-result timetable or not.

1. **Usability and Humanity Requirements:**

The system is intended for academic staff (of all majors) who may not be tech-savvy, so ease of use is essential. Thus, we made sure the software is user-friendly and easy to follow by adding the steps the user should follow on the main page.

1. **Look and Feel Requirements:**

The software has a frontend with aesthetics that mimic the BAU brand genre colors (dark blue, and white). This will enhance the user’s experience, presenting a sense of engagement and satisfaction.

1. **Operational and Environmental Requirements:**

The system performs well across various platforms and under real-world usage conditions. It is also responsive to multiple devices (Laptops, mobile devices…etc.) and work 24/7 with minimal energy usage.

1. **Cultural and Political Requirements:**

The system respects the cultural, institutional, and political environment of BAU University and provides the user with the opportunity to switch languages that they prefer.

1. **Legal Requirements:**

The system complies with legal standards and regulations governing data privacy, licensing, and institutional reporting. It is also important to note that the data collected is taken under the consent and agreement of the university and used only by authorized users.

# Design

## Class Diagrams

## Dynamic Model

## Subsystem Decomposition

## Hardware / software mapping

## User Interface

# Test Plans

Features to be tested / not to be tested

Pass/Fail Criteria

Approach

Suspension and resumption

Testing materials (hardware / software requirements)

Test cases

Testing schedule

# Implementation

Output

# Results Evaluation

# Conclusion

## Summary

## Novelty

## Integrity and Values

## Future Work

# References / Bibliography

1. *Alghamdi, H., Alsubait, T., Alhakami, H., & Baz, A. (2020). A review of optimization algorithms for university timetable scheduling. Engineering, Technology & Applied Science Research, 10, 6410–6417.* [*A Review of Optimization Algorithms for University Timetable Scheduling | Engineering, Technology & Applied Science Research*](https://www.etasr.com/index.php/ETASR/article/view/3832)
2. *Bashab, A., Ibrahim, A. O., Hashem, I. A. T., Aggarwal, K., Mukhlif, F., Ghaleb, F. A., & Abdelmaboud, A. (2023). Optimization techniques in university timetabling problem: Constraints, methodologies, benchmarks, and open issues. Computers, Materials & Continua, 74, 6461–6484.* [*CMC | Optimization Techniques in University Timetabling Problem: Constraints, Methodologies, Benchmarks, and Open Issues*](https://www.techscience.com/cmc/v74n3/50939)
3. *Birbas, T., Daskalaki, S., & Housos, E. (2009). School timetabling for quality student and teacher schedules. Journal of Scheduling, 12, 177–197.* [*School timetabling for quality student and teacher schedules | Journal of Scheduling*](https://link.springer.com/article/10.1007/s10951-008-0088-2)
4. *Chen, P., Sze, S. N., Goh, S. L., & Kendall, G. (2021). A survey of university course timetabling problem: Perspectives, trends and opportunities. IEEE Access, 9, 82031–82053.* [*A Survey of University Course Timetabling Problem: Perspectives, Trends and Opportunities | IEEE Journals & Magazine | IEEE Xplore*](https://ieeexplore.ieee.org/document/9499056)
5. *Abdipoor, S., Yaakob, R., Goh, S. L., & Abdullah, S. (2023). Meta-heuristic approaches for the university course timetabling problem. Intelligent Systems with Applications, 19, 200253.* [*Meta-heuristic approaches for the University Course Timetabling Problem - ScienceDirect*](https://www.sciencedirect.com/science/article/pii/S2667305323000789?via%3Dihub)

: Cite all ideas, concepts, text, data that are not your own. If you make a statement, back it up with your own data or a reference. All references cited in the text must be listed. There are two main ways to cite a reference within a text:

Citing the reference by author’s name: the author’s name must be placed at the end of the sentence that is taken from that reference along with the year of publication, then in the reference section the author’s name is to be arranged in alphabetical order.

Citing the reference by numbers: you should start numbering from 1 and continue according to order of appearance in text. Numbers should be placed the end of the sentence that is taken from that reference, then in the reference section you start your reference list from number 1.

You are recommended to use the APA writing style, which cites the reference by the author’s name, in your references’ citations.

The first line of each entry in your reference list should be on the left margin. Subsequent lines should be indented five spaces from the margin. All references should be double-spaced. Capitalize only the first word of a title or subtitle of a work. Italicize titles of books and journals. Note that the italicizing in these entries often continues

beneath commas and periods. Each entry is separated from the next by a double space (thus the entire reference list is double spaced, with no extra returns added).

Authors' names are inverted (last name first); give last name and initials for all authors of a particular work. Your reference list should be alphabetized by authors' last names. If you have more than one work by a particular author, order them by publication date, oldest to newest (thus a 1991 article would appear before a 1996 article). When an author appears as a sole author and as the first author of a group, list the one-author entries first. If no author is given for a particular source, alphabetize by the title of the piece and use a shortened version of the title for parenthetical citations. Use "&" instead of “and” on the reference page and only within parentheses when citing multiple authors of a single work in your text. At the end of the project list all references cited in the text in alphabetical order.

For an article in a journal:

***Author, A. A., Author, B. B., & Author, C. C. (Year of Publication). Title of article. Title of periodical, Volume Number, pages.***

Example 1: Harlow, H. F. (1983). Fundamentals for preparing psychology journal articles. Journal of Comparative and Physiological Psychology, 55, 893-896.

Example 2: Kernis, M. H., Cornell, D. P., Sun, C. R., Berry, A., & Harlow, T. (1993). There's more to self-esteem than whether it is high or low: The importance of stability of self-esteem. Journal of Personality and Social Psychology, 65, 1190-1204.

For a chapter in a book:

***Author, A. A., & Author, B. B. (Year of Publication). Title of chapter. In A. Editor &***

B. Editor (Eds.), Title of book (pages of chapter). Location: Publisher. When you list the pages of the chapter or essay in parentheses after the book title, use "pp." before the numbers: (pp. 1-21).

Example: O'Neil, J. M., & Egan, J. (1992). Men's and women's gender role journeys: Metaphor for healing, transition, and transformation. In B. R. Wainrib (Ed.), Gender issues across the life cycle (pp. 107-123). New York: Springer.

For a web page:

***Author, A. A., & Author, B. B. (Date of Publication or Revision). Title of full work [online]. Retrieved month, day, year, from source Web site: URL.***

Example: Chou, L., McClintock, R., Moretti, F. & Nix, D. H. (1993.) Technology and education: New wine in new bottles: Choosing pasts and imagining educational futures. Retrieved August 24, 2000, from Columbia University Institute for Learning Technologies Web site: <http://www.ilt.columbia.edu/publications/papers/newwine1.html>

For an online journal:

***Author, A. A., & Author, B. B. (Date of Publication). Title of article. Title of periodical, xx, xxx-xxx. Retrieved month, day, year, from URL.***

Example: Frederickson, B. L. (2000, March 7). Cultivating positive emotions to optimize health and well-being. Prevention &Treatment, 3 Article 001a. Retrieved November 20, 2000, from <http://journals.apa.org/prevention/volume3/pre0030001a.html>

# Appendix

Glossary

Naming Conventions and Definitions

Code and links

User Manual