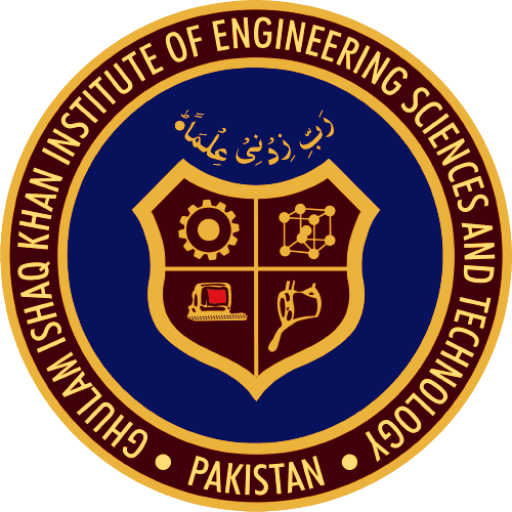
****

DSA-Project

100% Completion Report

# By: Muneeb Jawed 2023690

# Ibrahim Saqib 2023256

# Rafay Rauf 2023020

**Project Report: University Timetable System**

**1. Project Overview**

The **University Timetable System** is designed to help universities manage course schedules effectively by allowing students or administrators to add courses, select lecture halls, and generate a timetable for a given set of courses. This system allows flexibility in assigning courses to available slots and ensures that the timetable reflects the right course, instructor, lecture hall, and time slot.

The key functionalities include:

* **Course Addition**: Users can add a course by specifying its name, instructor, credit hours, lecture hall, and days.
* **Slot Selection**: Users can select the available time slots for each course.
* **Timetable Generation**: The system generates a timetable that displays the courses assigned to specific lecture halls and times.

**2. Problem Statement**

Universities often struggle with creating and managing timetables, especially when courses are numerous, and there are many conflicting schedules. The goal of this project is to automate the process of generating a timetable where courses are assigned to available lecture halls and time slots. This will save administrators time and help eliminate scheduling conflicts.

**3. System Architecture**

The system is built using a **client-server** architecture:

* **Frontend**: The frontend is a web-based interface where users can input course details and generate timetables. The frontend is developed using **HTML**, **CSS** (TailwindCSS), and **JavaScript**.
* **Backend**: The backend is a Flask-based web server that handles requests, processes course data, and generates the timetable. It is responsible for receiving course data from the frontend, processing it, and returning the generated timetable.
* **Database**: For this project, no persistent database is used. Instead, the course information is passed dynamically through JSON objects between the frontend and backend.

**4. Functional Requirements**

* **Course Information**: The user must provide the course name, instructor, credit hours, and lecture hall.
* **Days of the Week**: The user can select the days of the week the course will be held.
* **Time Slot Selection**: The user can select which time slots the course should take place in.
* **Timetable Generation**: After adding all the courses, the user can generate a timetable that shows the schedule in a grid format.

**5. Non-Functional Requirements**

* **Usability**: The system must be user-friendly and simple to use, ensuring a smooth experience for university administrators.
* **Scalability**: The system should be able to handle a large number of courses and generate timetables for multiple departments.
* **Performance**: The system should generate the timetable in real-time without significant delays.
* **Security**: The system should ensure that the data provided by users is securely processed and handled.

**6. System Components**

**Frontend:**

The frontend consists of the following components:

1. **Course Form**: A form where the user can input the course details such as course name, instructor, credit hours, lecture hall, days, and selected time slots.
2. **Timetable Display**: After generating the timetable, the system displays it in a grid format with time slots as rows and lecture halls as columns.
3. **Interactive Elements**: Users can interact with checkboxes to select the days and time slots for each course.

**Backend:**

The backend is responsible for:

1. **Processing Requests**: The backend receives course data (name, instructor, credit hours, lecture hall, days, and time slots) and stores it temporarily in memory.
2. **Timetable Generation**: Based on the provided data, the backend generates a timetable where courses are assigned to the appropriate lecture halls and time slots.
3. **Dynamic HTML Generation**: The backend generates the timetable in an HTML format, which is returned to the frontend for display.

**Timetable Grid:**

The timetable grid is structured as follows:

* **Rows**: Each row corresponds to a time slot (e.g., Slot 1: 8:00 AM - 9:00 AM).
* **Columns**: Each column corresponds to a lecture hall (e.g., LH1, LH2).
* **Slot Assignment**: If a course is assigned to a particular lecture hall and time slot, it will appear in that cell; otherwise, it will display "Free".

**7. Technologies Used**

* **Frontend**:
  + **HTML**: Used for the basic structure of the page.
  + **TailwindCSS**: Used for styling the page in a responsive and modern way.
  + **JavaScript**: Used to handle user interactions, course data management, and timetable generation.
* **Backend**:
  + **Flask**: A lightweight web framework used to handle HTTP requests, process data, and render HTML.
  + **Python**: Used to write the backend logic for generating the timetable.
* **Additional Libraries**:
  + **html2pdf.js**: A JavaScript library used to convert the generated timetable into a downloadable PDF file.

**8. Timetable Example**

For example, after entering the following courses:

1. **Course**: Math 101
   * **Instructor**: Dr. John
   * **Credit Hours**: 3
   * **Lecture Hall**: LH1
   * **Days**: Tuesday, Wednesday, Thursday
   * **Slots**: 2 (9:00 AM - 10:00 AM), 3 (10:00 AM - 11:00 AM)
2. **Course**: Physics 102
   * **Instructor**: Dr. Smith
   * **Credit Hours**: 3
   * **Lecture Hall**: LH2
   * **Days**: Monday, Wednesday
   * **Slots**: 1 (8:00 AM - 9:00 AM)

**Generated Timetable (Example):**

Day: Monday

------------------------------------------------------------------------------------------

Slot | LH1 | LH2 | LH3 | LH4 | LH5 |

------------------------------------------------------------------------------------------

Slot 1 | Free | Physics 102 (Dr. Smith) | Free | Free | Free |

Slot 2 | Free | Free | Free | Free | Free |

Slot 3 | Free | Free | Free | Free | Free |

Slot 4 | Free | Free | Free | Free | Free |

Slot 5 | Free | Free | Free | Free | Free |

Slot 6 | Free | Free | Free | Free | Free |

Slot 7 | Free | Free | Free | Free | Free |

Slot 8 | Free | Free | Free | Free | Free |

Day: Tuesday

-------------------------------------------------------------------------------------------------

Slot | LH1 | LH2 | LH3 | LH4 | LH5 |

-------------------------------------------------------------------------------------------------

Slot 1 | Free | Free | Free | Free | Free |

Slot 2 | Math 101 (Dr. John) | Free | Free | Free | Free |

Slot 3 | Math 101 (Dr. John) | Free | Free | Free | Free |

Slot 4 | Free | Free | Free | Free | Free |

Slot 5 | Free | Free | Free | Free | Free |

Slot 6 | Free | Free | Free | Free | Free |

Slot 7 | Free | Free | Free | Free | Free |

Slot 8 | Free | Free | Free | Free | Free |

**9. Challenges**

* **Time Slot Management**: Ensuring that courses fit into available time slots and lecture halls without conflicts was the main challenge. The system needed to handle conflicts between multiple courses trying to use the same slot or room.
* **User Interface**: Designing a simple, user-friendly interface to input data and generate the timetable dynamically.

**10. Future Enhancements**

* **Room Capacity Management**: The system could be extended to handle room capacity and prevent over-booking of large courses into small rooms.
* **Advanced Conflict Handling**: Add automatic conflict detection and resolution for courses that might overlap in time and room.
* **Integration with University Databases**: Allow the system to integrate with existing university databases to pull real course data (e.g., instructors, courses).

**11. Conclusion**

The **University Timetable System** successfully automates the scheduling process, making it easier for university administrators to organize and generate course timetables. The system provides a user-friendly interface and powerful backend logic to handle various scheduling requirements and slot assignments. The project can be expanded further to include additional features, such as conflict resolution, room capacity management, and integration with university data.

**Project Report Summary:**

This report provides an overview of the **University Timetable System**, explaining its functionality, architecture, and the technologies used to build it. It highlights the importance of automating the timetable generation process and outlines the project's current and potential future enhancements.