**University Timetable Generator - Project Report**

**NAME**: USMAN AHMED JAN-4744

MAZHAR ALI-4743

KHALID BADSHAZOI-4596

ABDUL NOOR-4739

**Submitted To:** [MR.SHAKEEL]  
**Department:** SOFTWARE ENGINEEERING

**Project Title:** University Timetable Generator

**Course:** Data Structures and Algorithms (DSA)  
**Date:** 23-June -2025

**1. Introduction**

The University Timetable Generator is a C++ command-line application designed to automatically generate class timetables for a university. It ensures no conflicts in teacher schedules and classroom timings using fundamental data structures and algorithms.

**2. Objectives**

* To apply core data structures such as arrays, vectors, maps, and sets.
* To implement scheduling logic using algorithmic techniques.
* To build an interactive CLI application.
* To enable exporting the schedule to a CSV file.

**3. Tools & Technologies**

* **Programming Language:** C++
* **Compiler:** GCC / Dev C++
* **Data Structures:**
  + Vectors: For storing lists of courses and schedules
  + Maps: For quick access to teacher's schedule
  + Sets: To avoid duplication of time slots

**4. OOP Design & Class Diagram**

* **Classes Used:**
  + Course: Contains course details.
  + ScheduledCourse: Holds scheduled information.
  + Timetable: Manages courses, scheduling, and display logic.

**UML Design Overview:**

+------------------+

| Course |

+------------------+

| - code |

| - name |

| - teacher |

| - hoursPerWeek |

+------------------+

+------------------+

| ScheduledCourse |

+------------------+

| - course |

| - day |

| - time |

+------------------+

+------------------+

| Timetable |

+------------------+

| - courses |

| - schedule |

| - teacherSlots |

+------------------+

| +addCourse() |

| +generateSchedule() |

| +displaySchedule() |

| +saveToCSV() |

| +inputCourses() |

+------------------+

**5. Implementation Summary**

* Input number of courses.
* For each course, enter name, code, teacher, and hours/week.
* Program schedules each course ensuring:
  + No teacher double-booking.
  + No time slot repetition.
* Outputs a well-formatted timetable.
* Allows exporting to timetable.csv.

**6. Algorithm Analysis**

* **Scheduling:**
  + Time complexity: O(n\*m), where n is courses and m is time slots.
  + Efficient use of sets/maps for collision checks.
* **Searching & Sorting:**
  + Sorting courses by name: O(n log n)
  + Searching course by code: O(n) linear scan

**7. Data Structure Justification**

* **Vector:** Dynamic storage of input/output data.
* **Map/Set:** Prevent schedule and teacher clashes quickly.
* **String Arrays:** Used for days and time slot labeling.

**8. Testing**

* Tested with multiple sets of input courses.
* Verified no overlapping slots or teacher assignments.
* Validated file export content.

**9. Optimizations**

* Separated logic into classes/methods.
* Allowed dynamic searching and sorting.
* Added data clearing and input validations.

**10. Conclusion**

This project demonstrates strong application of object-oriented programming and data structures. It is a practical solution for educational institutions to automate timetable generation.

**11. Future Work**

* GUI version using Qt or SFML
* Integration with student/course database
* Web-based deployment