**SCHOOL TIMETABLE GENERATOR**

**Project report submitted in partial fulfillment of the Requirements for theAward of the Degree of**

**BACHELOR OF TECHNOLOGY**

**In**

**COMPUTER SCIENCE AND ENGINEERING**

**By**

**POORNA TEJA-24KB1A06P0**

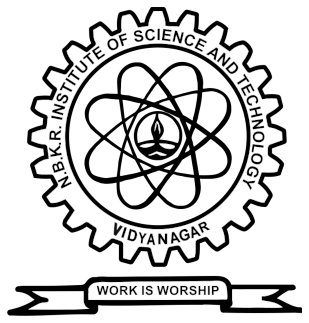
**Mouli-24KB1A05NF**

**VENKATESH-24KB1A05NK**

**SAI CHARAN-24KB1A05PD**

**Under the Guidance of**

**Ashok Selva Kumar**

****

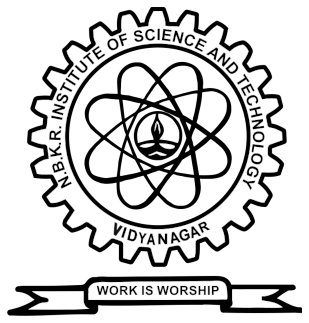
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**NBKRIST**

NBKR INSTITUTE OF SCIENCE AND TECHNOLOGY

**(AUTONOMOUS)**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**



###### CERTIFICATE

This is to certify that the project report entitled YOUR PROJECT TITLE being submitted by

POORNA TEJA-24KB1A06P0

Mouli-24KB1A05NF

VENKATESH-24KB1A05NK

SAI CHARAN-24KB1A05PD

in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer Science and Engineering to the NBKR INSTITUTE OF SCIENCE AND TECHNOLOGYis a record of bonafied work carried out under my guidance and supervision.

| **SMT.Geetha Reddy**  **Assistant Professor** | **Dr A Raja Sekhar Reddy**  **M.Tech, Ph.D**  **Head of the Department** |
| --- | --- |

**DECLARATION**

I hereby declare that the dissertation entitled

SCHOOL TIMETABLE GENERATOR submitted for the B.Tech Degree is my original work and the dissertation has not formed the basis for the award of any degree, associateship, fellowship or any other similar titles.

Place: Date:

POORNA TEJA

**24KB1A05P0**

Acknowledgement

I would like to express my sincere gratitude to all those who helped me complete this project titled **"SCHOOL TIMETABLE GENERATOR"**.

First and foremost, I would like to thank my faculty guide, **Geetha Reddy** ,for their continuous support, valuable feedback, and encouragement throughout the development of this project.

I would also like to thank the Head of Department, **Dr A Raja Sekhar Reddy** ,for providing us with the necessary resources and environment to carry out this work effectively.

A special thanks to my teammates and friends for their constant motivation, brainstorming sessions, and cooperation, which made this learning experience memorable and enriching.

Finally, I am grateful to my family for their endless support and encouragement, which kept me motivated throughout this project.

This project has helped me enhance my technical knowledge in **C Programming, Linked Lists, Structures, and Dynamic Memory Management**, and it has given me a glimpse of real-world application development challenges.

**Thank you all once again!**

Abstract of the Project.

INFORMATION…….

# **1. Introduction**

## 1.1 Introduction

The **School Timetable Generator** is a console-based application developed in the **C programming language** to manage seat reservations for a bus. The goal of this project is to simulate a real-world booking system **.**

**1.2** **Context**

Creating an efficient and balanced timetable is a key administrative task in any educational institution. This C program simplifies the process by allowing users to input a list of subject-teacher combinations and automatically generate a basic weekly schedule.

The program distributes classes across 5 weekdays and 5 fixed time slots using a round-robin allocation method. It ensures fair distribution and provides a clear textual timetable that can be easily reviewed or modified. This project demonstrates practical use of arrays, string handling, and nested loops in C programming, making it both a functional tool and an excellent educational exercise in basic scheduling logic

## 1.3 Scope

* View the seat map showing availability and bookings.
* Book available seats by selecting row and column numbers.
* Cancel existing bookings.
* Save current bookings to a text file for record-keeping.

## 1.4 Objectives

* Automate Scheduling
* Eliminate the need for manual timetable creation, reducing human effort and errors.
* Efficient Resource Allocation
* Assign teachers, classrooms, and other resources without conflicts.
* Conflict-Free Timetabling
* Ensure no teacher, student group, or classroom is scheduled in two places at the same time.
* Flexibility and Customization
* Support specific constraints like teacher availability, subject preferences, and class durations.
* Time Optimization
* Minimize idle time for teachers and students, optimizing teaching hours.
* User-Friendly Interface
* Allow easy input of data and visualization of the generated timetable.
* Scalability
* Handle large numbers of classes, teachers, and subjects without performance degradation.
* Adaptability
* Support quick changes and re-generation of timetables if schedules change.

ALGORITHM

1. Start

2. Prompt the user to enter the number of subject-teacher pairs

3. Store each subject-teacher entry in an array

4. Initialize a timetable matrix for 5 days and 5 time slots

5. Using a round-robin method:

1. Assign each subject teacher entry to the timetable slots day by day

6. Display the timetable:

1. Show days as rows and time slots as columns

2. Print subject-teacher assignments in each cell

7. End

# **6. Coding**

This system mainly uses:

* **C Programming Language – for writing the entire logic and user interface.**
* 2D Arrays – to store and manage the seat layout (availability/bookings).
* Conditional Statements – to control booking and cancellation logic.
* Loops – for displaying and updating the seat map.
* Functions – for modular programming (e.g., bookSeat(), cancelSeat()).
* File Handling – to save seat bookings to a text file (bookings.txt).
* Console I/O – for user interaction through terminal (e.g., scanf, printf).

**Conclusion:**

The School Timetable Generator is a practical C-based solution for automating weekly class schedules. By allowing user-defined subject-teacher entries and distributing them across predefined time slots, it streamlines a traditionally manual process. While it has limitations such as lack of conflict management and data storage, it effectively demonstrates key programming concepts like arrays, loops, and string handling. This program serves as a solid foundation for building more advanced and feature-rich scheduling systems in the future.

**References:**

* The following format used for writing the References.

1. Algorithms and Data Structures: The Basic Toolbox by Kurt Mehlhorn and Peter Sanders.

2. C Data Structures and Algorithms by Alfred V. Aho, Jeffrey D. Ullman, and John E. Hopcroft.

3. Problem Solving with Algorithms and Data Structures by Brad Miller and David Ranum.

4. Introduction to Algorithms by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.

5. Algorithms in C, Parts 1-5 (Bundle): Fundamentals, Data Structures, Sorting, Searching, and Graph Algorithms by Robert Sedgewick.