**UNIVERSITY INSTITUTE OF COMPUTING**

**PROJECT REPORT**

**ON**

**CALENDAR**

Program Name: BCA

Subject Name/Code: Data Structures(23CAT-201)

**Submitted by: Submitted to:**

**Name:** Karan Kulraj Singh **Name:** Mrs. Shilpi Mittal

**UID:** 23BCA10783 **Designation:** Co-ordinator

**Section:** BCA – **3** “A”

**Group:** **6**

ABSTRACT

Introduction:

This project aims to develop a calendar generation program, which displays the calendar for any given month and year. The application utilizes Zeller’s Congruence algorithm to accurately determine the starting day of the month and handles leap years to ensure precise calculations.

The program takes user input for the year, then prints a well-formatted calendar with the days aligned correctly under their respective weekdays. This project demonstrates the application of fundamental control structures, date algorithms, and formatting techniques in C++. It serves as a practical example of building a utility tool, reinforcing concepts such as modular programming, input handling, and basic arithmetic algorithms.

Technique:

The application is developed using C++ with a focus on object-oriented programming (OOP) principles. It employs the following techniques:

1. **Modular Programming:** The code is divided into smaller functions ( dayNumber, getMonthName, numberOfDays, printCalendar) to improve readability, maintainability, and reusability.
2. **Algorithm Implementation (Zeller’s Congruence):** Zeller’s Congruence is used to calculate the starting day of the week for the 1st of the given month. This ensures accurate alignment of days within the calendar.
3. **Conditional Logic and Leap Year Handling:** Conditional statements are used to determine whether the year is a leap year and to assign the correct number of days to February.
4. **Mathematical Operations:** Arithmetic operations and modulus (%) are employed to determine the placement of days and ensure correct breaks after Saturdays.

System Configuration:

* **OS:** Windows 10 or MACOSX
* **Processor:** Intel Core i3 (minimum); Core i5 or higher recommended
* **RAM:** 4 GB (minimum); 8 GB recommended
* **Development Environment:** Any C++ IDE (e.g., Visual Studio, Code::Blocks) or Visual Studio Code with a C++ compiler (GCC or Microsoft C++ Compiler)

SUMMARY

Input:

In this project, the Calendar Generator takes the following inputs:

1. Year (e.g., 2024):

• The user provides the year to generate the calendar for that specific year.

• This input is crucial for correctly handling leap years (e.g., February has 29 days in a leap year).

Output:

Output:

• The program generates and displays a monthly calendar based on the given year.

• The days are aligned correctly under the respective weekday headings (Sunday to Saturday).

• For leap years, February will display 29 days; otherwise, it will show 28 days.

