A Course-Based Project Report on

CALENDAR

Submitted to the

Department of Electronics and Instrumentation

in partial fulfilment of the requirements for the completion of course

PROGRAMMING FOR PROBLEM SOLVING LABORATORY (22ES1CS101)

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND INSTRUMENTATION ENGINEERING

Submitted by

23071A1021	K Dhanush
23071A1022	D Pranavanath
23071A1023	D Varshitha
23071A1024	E Eshwithasri
23071A1025	G Diskshitha

Under the guidance of

Mrs. SRAVANTHI

Assistant Professor, Department of IT, VNRVJIET

DEPARTMENT OF ELECTRONICS AND INSTRUMENTATION ENGINEERING

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING & TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade, NBA

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

DECEMBER 2023

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade, NBA Accredited for CE, EEE, ME, ECE, CSE, EIE, IT B. Tech Courses, Approved by AICTE, New Delhi, Affiliated to JNTUH, Recognized as "College with Potential for Excellence" by UGC, ISO 9001:2015 Certified, QS I GUAGE Diamond Rated

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet(SO), Hyderabad-500090, TS, India

DEPARTMENT OF ELECTRONICS AND INSTRUMENTATION ENGINEERING

CERTIFICATE

This is to certify that the project report entitled **"Calendar"** is a bonafide work done under our supervision and is being submitted by**K.Dhanush(23071A1021), D.Pranavanath(23071A1022),DVarshitha(23071A1023,E.Eswithasri(23071A1024),G.Dikshitha(23071A1025)** in partial fulfilment for the award of the degree of **Bachelor of Technology** in Electronics and Instrumentation, of the VNRVJIET, Hyderabad during the academic year 2022-2023.

Mrs. K Sravanthi
Assistant Professor
Department of IT

Dr R Manjula Sri
Professor & HOD
Department of EIE

Course based Projects Reviewer

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous Institute, NAAC Accredited with 'A++' Grade,

Vignana Jyothi Nagar, Pragathi Nagar, Nizampet(SO), Hyderabad-500090, TS, India

DEPARTMENT OF INFORMATION TECHNOLOGY

DECLARATION

We declare that the course based project work entitled "**Calendar**" submitted in the Department of Electronics and Instrumentation, Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology, Hyderabad, in partial fulfilment of the requirement for the award of the degree of **Bachelor of Technology in Electronics and Instrumentation** is a bonafide record of our own work carried out under the supervision of **Sravanthi, Assistant Professor, Department of IT, VNRVJIET.** Also, we declare that the matter embodied in this thesis has not been submitted by us in full or in any part thereof for the award of any degree/diploma of any other institution or university previously.

Date:30.12.2023

Place: Hyderabad.

K Dhanush	D Pranavath	D Varshitha
(23071A1021)	(23071A1022)	(23071A1023)

E Eshwaritha	G Diskshitha
(23071A1024)	(23071A1025)

ACKNOWLEDGEMENT

We express our deep sense of gratitude to our beloved President, Sri. D. Suresh Babu, VNR Vignana Jyothi Institute of Engineering & Technology for the valuable guidance and for permitting us to carry out this project.

With immense pleasure, we record our deep sense of gratitude to our beloved Principal, Dr. C.D Naidu, for permitting us to carry out this project.

We express our deep sense of gratitude to our beloved Professor Dr. R Manjula sri, Professor and Head, Department of Electronics & Instrumentation Engineering VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad-500090 for the valuable guidance and suggestions, keen interest and through encouragement extended throughout the period of project work.

We take immense pleasure to express our deep sense of gratitude to our beloved Guide, **Mrs. Sravanthi**, Assistant Professor of Information Technology, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad, for his/her valuable suggestions and rare insights, for constant source of encouragement and inspiration throughout my project work.

We express our thanks to all those who contributed for the successful completion of our project work.

23071A1021	K Dhanush
23071A1022	D Pranavanath
23071A1023	D Varshitha
23071A1024	E Eshwithasri
23071A1025	G Diskshitha

ABSTRACT

This C programming project aims to create a calendar printing program. The program will prompt users to input the desired year and month, and then generate a calendar for that specific month. It will utilize basic C programming concepts such as loops and conditionals to organize and display the days of the week and corresponding dates. The project's objective is to enhance the user's understanding of fundamental programming constructs while providing a practical application for managing and presenting calendar information.

The calendar printing program will include features such as highlighting the current day, accommodating leap years, and ensuring proper alignment of days within the weeks. The user-friendly interface will guide users through the input process, allowing them to easily navigate and view calendars for different months and years. The code will be structured to promote modularity and readability, encouraging a learning experience for those new to C programming. This project aims to reinforce skills in handling user input, managing data structures, and implementing logic for date-related computations.

Additionally, the program will implement error handling to address invalid inputs, guiding users to provide accurate information for year and month selections. It will employ standard C libraries for date and time functions to facilitate accurate calendar generation. The code will emphasize clarity and adherence to best practices, serving as an educational resource for those seeking to improve their programming skills. By completing this project, users will gain practical experience in building a functional and interactive C program while creating a useful tool for displaying monthly calendars.

1.Table of Contents

CHAPTER-1

INTRODUCTION

1.1 PROBLEM DEFINITION

- Problem Definition for C Program on Calendar:

Design a C program to implement a calendar application that allows users to perform various operations related to dates and months. The program should provide the following functionalities:

1. Display the current month with days organized in a calendar format.
2. Allow the user to navigate to the previous or next month.
3. Enable the user to input a specific date and validate its correctness.
4. Determine the day of the week for a given date.
5. Find the number of days between two given dates.
6. Provide an option to set reminders for specific dates.

Ensure that the program handles leap years, invalid inputs, and provides a user-friendly interface for seamless interaction. The goal is to create a reliable and efficient calendar program that meets the needs of users managing dates and events.

1.2 OBJECTIVES

Objectives for C Program on Calendar:

1.”Display Current Month:**

- Implement a function to display the current month with days organized in a calendar format.

2.**Navigation:**

- Allow users to navigate to the previous or next month, providing a seamless way to explore different months.

3.**Input Validation:**

- Enable the user to input a specific date and implement robust validation to ensure the correctness of the entered date.

4.**Day of the Week:**

- Implement a feature to determine the day of the week for a given date, aiding users in understanding the calendar layout.

5.**Leap Year Handling:**

- Ensure proper handling of leap years to maintain accurate date calculations.

6.**Days Between Dates:**

- Develop a function to calculate the number of days between two given dates, facilitating date range calculations.

7.**Reminder System:**

- Provide an option for users to set reminders for specific dates, creating a basic event management system.

8.**User-Friendly Interface:**

- Design an intuitive and user-friendly interface for easy interaction, enhancing the overall user experience.

9.**Error Handling:**

- Implement robust error-handling mechanisms to gracefully manage unexpected inputs or errors during program execution.

10.**Efficiency:**

- Optimize the program for efficiency in terms of memory usage and execution speed, ensuring a responsive application.

11.**Documentation:**

- Include clear and concise documentation within the code to aid developers in understanding the program's structure and functionality.

12.**Testing:**

- Conduct thorough testing to identify and rectify any bugs or issues, ensuring the program's reliability and accuracy.

1.User Interface:

i. You could prompt the user to enter a month and year.

ii. Then display the corresponding calendar.

2.Result Display:

i. If you run the provided C code, it will display the current date in the format MM/DD/YYYY.

ii. To explore more functions and customize the code to suit your specific needs..

3.Error Handlin: This code checks if localtime returns a null pointer, and if so, it prints an error message using perror and returns a non-zero value to indicate an error.

4.Modularity and Readability:

i.Design the program in a modular way for easy maintenance and scalability.

ii.Use clear and descriptive variable and function names to enhance code readability.

5.Documentation:

iii. Include comments and documentation to explain the functionality of different parts of the code.

iv. Provide instructions for users on how to use the program.

6.Testing:

To start with defining the structure of a date, handling input for the month and year, and then displaying the calendar.

CHAPTER-2

SOURCE CODE

```
#include <stdio.h>
// Function to determine the day of the week for a given date int dayOfWeek(int year,

int month, int day)

{

if (month < 3) {
month += 12; year--;

}

int h = (day + 2 * month + 3 * (month + 1) / 5 + year + year / 4 - year / 100 + year / 400)

% 7; return h;

}

// Function to print the calendar for a given month and year void printCalendar(int

year, int month) {

int daysInMonth[] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

int totalDays, startingDay, i, j;

// Check for leap year

if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) { daysInMonth[2] = 29; //

February has 29 days in a leap year

}
totalDays = daysInMonth[month]; startingDay = dayOfWeek(year, month, 1);

// Print the header

printf("\n                    \n");

printf("   Calendar %d - %d\n", month, year); printf("  \n");
```

```
printf(" Sun Mon Tue Wed Thu Fri Sat\n");
// Print leading spaces

for (i = 0; i < startingDay; i++)
{
printf("                                                    ");
}

// Print days
for (j = 1; j <= totalDays; j++) { printf("%4d", j);

if ((i + j) % 7 == 0) {

printf("\n");

}

}
printf("\n\n");
}

int main() {  int year, month;

// Input year and month from the user printf("Enter year: "); scanf("%d", &year);

printf("Enter month (1-12): "); scanf("%d", &month);

// Validate input
if (month < 1 || month > 12) {
printf("Invalid month. Please enter a month between 1 and 12.\n"); return 1;

}

// Print#include <stdio.h>


// Function to determine the day of the week for a given date int dayOfWeek(int year,

int

month, int day)

{

if (month < 3) { month += 12;

year--;

}
```

```
int h = (day + 2 * month + 3 * (month + 1) / 5 + year + year / 4 - year / 100 + year / 400)

% 7; return h;

}
// Function to print the calendar for a given month and year void printCalendar(int

year, int month)

{

int daysInMonth[] = {0, 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};

int totalDays, startingDay, i, j;

// Check for leap year

if ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0)) { daysInMonth[2] = 29; //

February has 29 days in a leap year

}
totalDays = daysInMonth[month]; startingDay = dayOfWeek(year, month, 1);

// Print the header

printf("\n        \n");

printf("   Calendar %d - %d\n", month, year); printf("           \n");

printf(" Sun Mon Tue Wed Thu Fri Sat\n");

// Print leading spaces
for (i = 0; i < startingDay; i++) {
printf("                                                  ");

}
// Print days
for (j = 1; j <= totalDays; j++) { printf("%4d", j);

if ((i + j) % 7 == 0) {

printf("\n");


}

}
```

```
printf("\n\n");

}
int main() {  int year, month;

// Input year and month from the user printf("Enter year: "); scanf("%d", &year);

printf("Enter month (1-12): "); scanf("%d", &month);

// Validate input

if (month < 1 || month > 12) {

printf("Invalid month. Please enter a month between 1 and 12.\n"); return 1;

}

// Print the calendar printCalendar(year, month);

return 0;

} the calendar printCalendar(year, month);

return 0;

}
```

CHAPTER-3

TEST CASES/ OUTPUT

********CALENDAR*********

Enter year: 2023
Enter month (1-12): 12

```
--------------------------
  Calendar 12 - 2023
--------------------------
```

Sun	Mon	Tue	Wed	Thu	Fri	Sat
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31						

CHAPTER-4

CONCLUSION

In conclusion, the provided C program demonstrates a simple console-based calendar interface. It allows users to input a month and year, and then displays the corresponding calendar. Keep in mind that this is a basic example, and for more advanced user interfaces or graphical representations, you would need to explore graphical libraries compatible with C, such as GTK or Qt, depending on your project requirements.

The C programming example provided offers a basic console-based calendar interface. Users can input a month and year, and the program displays the corresponding calendar for that month. This serves as a starting point for understanding how to structure calendar-related logic in C. For more advanced features
or graphical interfaces, exploring additional libraries or incorporating user input validation could enhance the program's.

REFERENCES

- Programming through C-BALAGURU SWAMY
- We would like to express our deep gratitude for the invaluable guidance from diverse references that enriched the completion of our project