PURBANCHAL UNIVERSITY

KULESHWOR AWAS CAMPUS

FIRST SEMESTER PROJECT

ON

“CALENDAR MANAGEMENT SYSTEM”

In the partial fulfillments for the requirement of the first Semester Project-I (Subject code-BIT106CO) in the completion of Bachelor of Information Technology (BIT) degree at Kuleshwor Awas Campus, under Purbanchal University.

Submitted By:

Name: Pasang Dorje Tamang

S.N: 312113

Name: Prasanna Katuwal

S.N: 312114

Submitted To:

Purbanchal University

STUDENT'S DECLARATION

We following students, hereby declare that the Project Report is titled

"Calendar Management System" is a result of our own work and our indebtedness to other work publications, references, if any, have been dully acknowledged. If we are found guilty of copying any other report or published information and showing as our original work, we understand that we shall be liable and punishable by Purbanchal University, which may include fail in examination, 'Repeat study and re-submission of the report' or any other punishment the Purbanchal University may decide.

We further certify that this Project submitted in partial fulfillment of the requirement for the award of Bachelor in Information Technology (BIT) of the Purbanchal University is our original work and has not been submitted for award of any other degree or other similar title or prizes.

S.N	Name	Redg	Roll no.	Signature
1.	Pasang Dorje Tamang	177-3-2-01980-2024	312113	
2.	Prasanna Katuwal	177-3-2-01981-2024	312114	

Purbanchal University
Office of the Examination Management, Biratnagar

EXAMINER'S CERTIFICATION

This is to certify that the project entitled "CALENDAR MANAGEMENT SYSTEM" has been successfully completed by Mr. Pasang Dorje Tamang (S/N: 312113) & Mr. Prasanna Katuwal (S/N:312114) in partial fulfillment of Degree of Bachelor of Information Technology of Purbanchal University during the academic year 2016 Under the guidance of Mr. Narayan G.C.

…………………………………………………….
Department of Science and Technology
Kuleshwor Awas Campus
PURBANCHAL UNIVERSITY

TO WHOM IT MAY CONCERN

This is to clarify that Mr. Pasang Dorje Tamang and Mr.Prasanna Katuwal, students of Bachelor in information technology (BIT) program, have successfully completed their first-semester project named “Calendar Management System”.
This project is the original work of Mr.Pasang Dorje Tamang and Mr.Prasanna Katuwal, carried out under the supervision of Mr. Narayan G.C. as per guideline provided by Purbanchal University, and is certified as per the student’s declaration that the project “Calendar Management System” has not been submitted or presented Elsewhere as part of any academic work.

The details of the student are as follows:-

Name of student: -Pasang Dorje Tamang, Prasanna Katuwal

Course / Semester: - BIT / first Semester

Subject: - Project-I

Subject Code: - BIT106CO

……………………………………..

Narayan G.C.
Program Director, BIT
KULESHWOR AWAS CAMPUS

ABSTRACT

The Calendar Management System is a console-based application developed in the C programming language that facilitates efficient personal scheduling and event management. This system enables users to view monthly calendars, manage events, and maintain a to-do list. It offers core functionalities such as viewing a calendar for a specified month and year, adding and deleting events, and displaying stored events in a readable format. Additionally, the integrated to-do list module allows users to add tasks, mark them as completed, and review pending or completed tasks.

The calendar view takes leap years into account and computes the correct day of the week for the beginning of each month, ensuring accurate calendar representation. Events are stored in a text file (event.txt), making the system simple yet persistent across sessions. The to-do list operates similarly, storing active tasks in todo.txt and completed tasks in complete.txt.

Designed with simplicity and utility in mind, the Calendar Management System offers a practical solution for students and professionals to organize their time and activities without needing complex software. Its file-based architecture ensures data persistence while maintaining a lightweight footprint. This project demonstrates fundamental programming concepts such as file handling, modular function design, and user input validation, and basic date computations.

ACKNOWLEDGEMENT

We take this opportunity to express our profound appreciation and unfathomable regards to the Information Technology (IT) department for this commendable guidance, monitoring and constant encouragement throughout the course of this project. The help and guidance given by shall carry us a long way, in the journey on which we are about to embark.

We also take this opportunity to express a deep sense of gratefulness to Mr. Narayan GC for his amiable support, valuable information and guidance, which helped us in completing this task throughout its various stages. We also want to thank our program director Mr. Narayan G.C. who helped us a lot during the complete project by giving us his precious time. We are also indebted to all members of Kuleshwor Awas Campus, for the help provided by them in their in their respective fields. We are grateful for their cooperation during the period of our project.

Finally we would also like to express lots of thanks to PURBANCHAL UNIVERSITY for designing such a wonderful course structure. It will help us to get more knowledge in the field of Information Technology & help us to have a bright future in the field of technology.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

BIT	Bachelor of Information Technology
C	(Programming Language) C
GCC	GNU Compiler Collection
IDE	Integrated Development Environment
PC	Personal Computer
S/N	Serial Number
UI	User Interface

CHAPTER 1: INTRODUCTION

1.1 Introduction

Time management plays a crucial role in both personal and professional life. As our daily tasks, responsibilities, and deadlines continue to grow, having a well-organized scheduling system has become more important than ever. A Calendar Management System helps users keep track of important dates, plan events, and manage tasks efficiently.

Traditionally, people rely on notebooks or basic reminders, which can be unreliable and hard to maintain over time. Our project, "Calendar Management System", aims to provide a simple yet effective digital solution to these problems. It allows users to view monthly calendars, add and manage events, and maintain a to-do list with the ability to mark tasks as completed.

This system is especially helpful for students, professionals, and anyone looking to organize their daily life more efficiently without relying on external applications. It reduces the chances of missing important dates and helps users stay productive and on schedule.

1.2 Objective Of This Project

Objective refers to the specific goal or purpose of a project. It describes what the project aims to achieve**.**

- To provide a simple and efficient way for users to manage dates, events, and daily tasks.
- To help users schedule and view events on specific dates with ease.
- To improve time management by allowing users to maintain and track a personal to-do list.
- To reduce the chances of missing important dates or tasks.
- To minimize the use of paper for planning and organizing schedules.

- To ensure faster access to past and upcoming events and tasks through a user-friendly interface.
- To avoid data redundancy by storing events and tasks in organized digital format using file handling.

1.3 Methodology

The development of the Calendar Management System followed a modular and structured approach aimed at creating a lightweight, user-friendly, and efficient scheduling tool using the C programming language.

The first phase involved requirement analysis and planning, where essential features like calendar display, event management, and to-do list tracking were identified. Decisions regarding file-based storage for persistence and a console-based user interface were finalized to ensure simplicity and portability.

In the second phase, design and implementation, the application was broken down into modular C functions for each core feature. Special focus was given to accurate calendar generation using leap year logic, file handling for storing events and tasks, and validating user inputs for correctness and reliability.

The final phase was testing and refinement, where individual modules were tested for edge cases such as invalid dates, duplicate entries, and file access errors. Based on test results and user experience, improvements like clear prompts, input confirmations, and structured menus were added to enhance usability and system stability.

This step-by-step methodology ensured the system was robust, practical, and met the intended goals for time and task management.

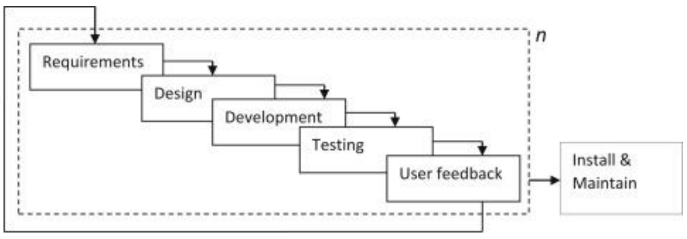


Figure 1.3 Incremental Approach

1.4 Software / Language Used

C is a powerful general-purpose programming language developed in the early 1970s by Dennis Ritchie at Bell Labs. It is widely used for system programming and application development due to its efficiency and ability to work closely with hardware. C is considered a middle-level language because it combines features of both high-level and low-level programming. Its simplicity and speed make it well-suited for developing software like this calendar management system, which involves handling dates, events, and task management efficiently.

1.5 Time Schedule

The project was developed over a period of 5 to 6 weeks and was divided into clear, structured phases to ensure smooth progress. The timeline outlines each stage of development, starting from idea selection and requirement analysis, moving through coding and implementation of features like the calendar, event manager, and to-do list, and ending with testing, documentation, and final submission. This organized progression helped maintain focus, track milestones, and ensure that each module was completed effectively before moving on to the next.

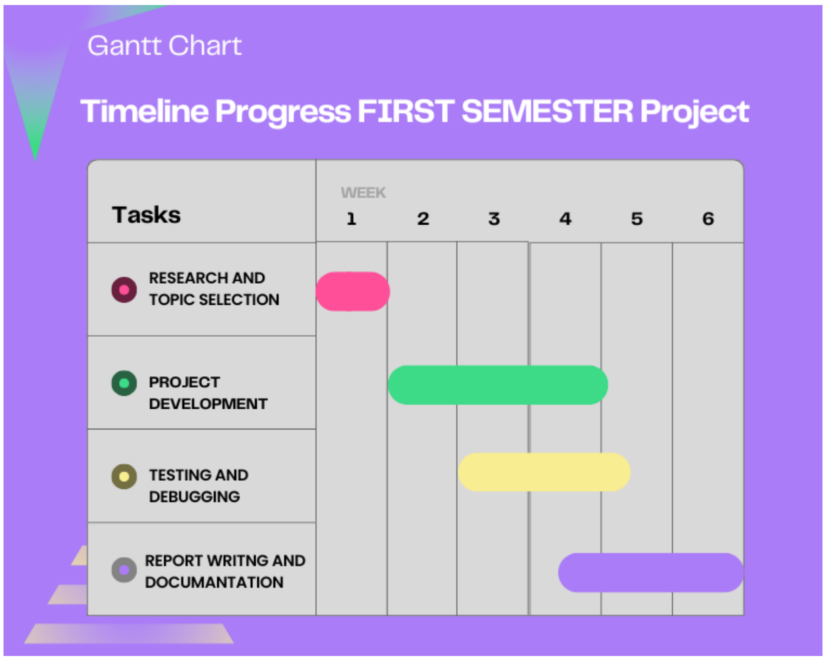


Figure 1.5 Gantt chart

Chapter 2: SYSTEM REQUIREMENT

2.1 Hardware

To ensure smooth development and execution of the project, the following hardware components were required and used during the implementation phase.

- PC with Pentium II Processor (260 MHz) or higher
- 32 MB RAM or more
- Color Monitor (preferred)
- Hard disk with at least 50 MB of free space

2.2 Software

The successful development and execution of the project relied on specific software tools and environments that supported coding, compilation, and testing of the system.

- The project requires a C compiler such as GCC or Turbo C to compile and execute the source code.
- An Integrated Development Environment (IDE) like Code::Blocks, Dev-C++, or Visual Studio Code can be used to write and manage the code more efficiently.
- The system is compatible with Windows, Linux, or macOS operating systems.
- A text editor is needed to view and modify output files such as event.txt and todo.txt.

Chapter 3: SYSTEM DESIGN

3.1 Introduction to the Project

This project is a Calendar and To-Do List Management System developed in the C programming language. It provides users with a simple and efficient way to view monthly calendars, manage events by adding, viewing, and deleting them, and organize daily tasks through a to-do list with options to mark tasks as completed. The system ensures data persistence using file handling and offers a user-friendly menu-driven interface for easy navigation.

3.2 Algorithm

The algorithm for the Calendar and To-Do List Management System is designed to handle date calculations, event management, and task tracking efficiently. It starts by calculating the total number of days to determine the day of the week for any given date, taking leap years into account. For event management, the system allows users to add, view, and delete events while ensuring that past dates cannot be selected. The to-do list module supports adding tasks, viewing pending and completed tasks, and marking tasks as complete. File handling is used throughout to store and retrieve data, ensuring persistence across sessions. The algorithm is implemented through modular functions to maintain clarity and ease of debugging.

The system consist of different sets of algorithms:

Step 1: Display main menu with options:

- View Calendar
- Add Event
- View Event
- Delete Event
- To-Do List
- Exit Program

Step 2: Wait for user input (1–6) and process based on selected option.
Step 3: If Option 1 (View Calendar):

- Ask for year and month.
- Calculate first weekday of the month.
- Display calendar for selected month.

Step 4: If Option 2 (Add Event):

- Ask for event date.
- Validate that it’s a future date.
- Ask for event description.
- Save to event.txt.

Step 5: If Option 3 (View Event):

- Read and display all events from event.txt.
- If none, show “No event found”.

Step 6: If Option 4 (Delete Event):

- Show existing events.
- Ask for date to delete.
- Remove matching entry from file.

Step 7: If Option 5 (To-Do List):
Display submenu:

1. Add To-Do
2. View To-Do
3. Mark as Completed
4. View Completed
5. Return to Main Menu

Each sub-option performs corresponding file operations (todo.txt, complete.txt).

Step 8: If Option 6 (Exit):

- Terminate program using return 0.

Step 9: Stop

3.3 Flowchart

The flowchart represents the logical structure and sequence of operations in the Calendar and To-Do List Management System. It begins with the main menu, allowing users to choose between viewing the calendar, managing events, or accessing the to-do list. Each option leads to a specific set of operations such as adding, viewing, or deleting data. Decision-making blocks guide the program flow based on user input, while arrows indicate the transition between different processes.

The flowchart helps visualize the overall functionality and makes it easier to understand the system's working.

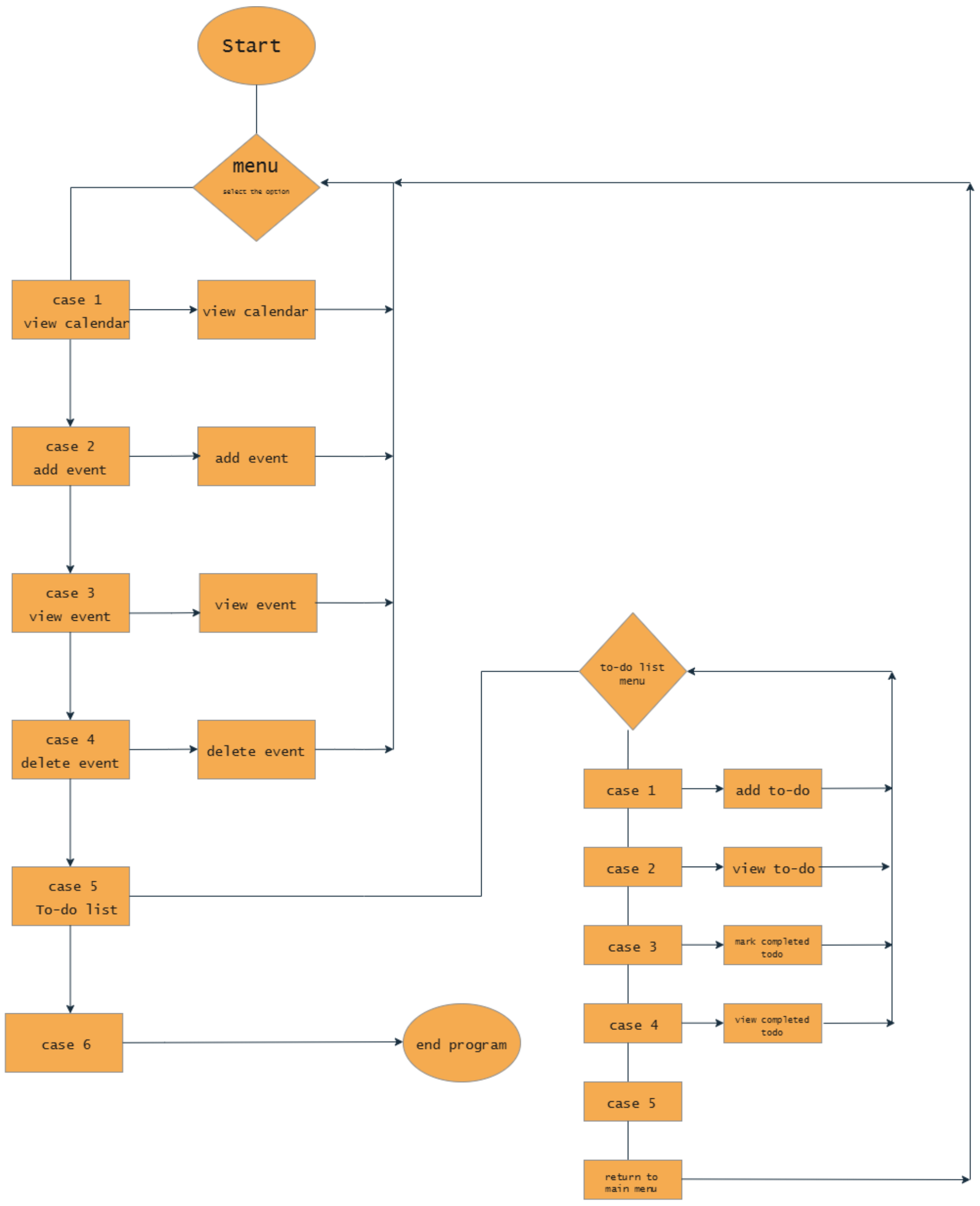


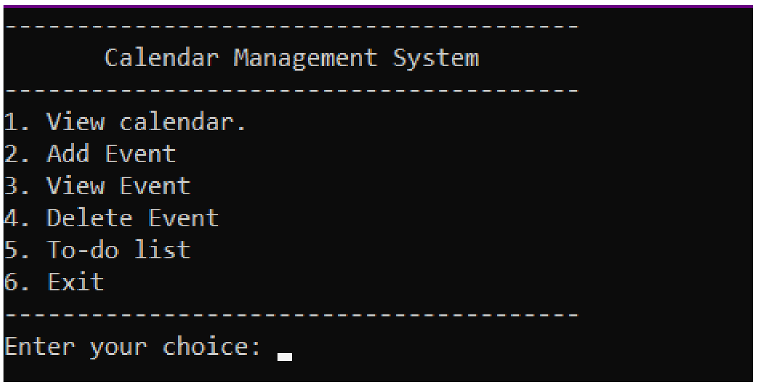
Figure 3.3 Flowchart

3.4 System Requirement

3.4.1 Functional Requirement

3.4.1.1 Menu

The system provides a user-friendly, menu-driven interface that allows users to navigate between key functionalities such as viewing the calendar, managing events, and handling to-do lists. Each menu option leads to a specific set of features designed for ease of access and interaction.

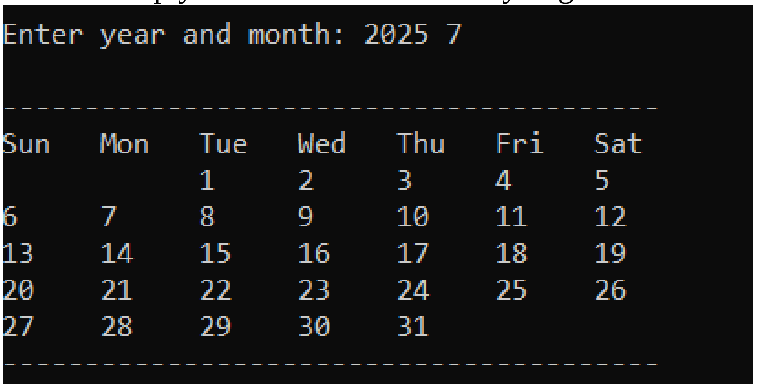


```
----------------------------------------
       Calendar Management System
----------------------------------------
1. View calendar.
2. Add Event
3. View Event
4. Delete Event
5. To-do list
6. Exit
----------------------------------------
Enter your choice: _
```

Figure 3.4.1.A Main Menu

3.4.1.2 View Calendar

The view calendar feature allows the user to display the calendar of any specified month and year. It accurately shows the layout of days, taking into account leap years and correct weekday alignment.

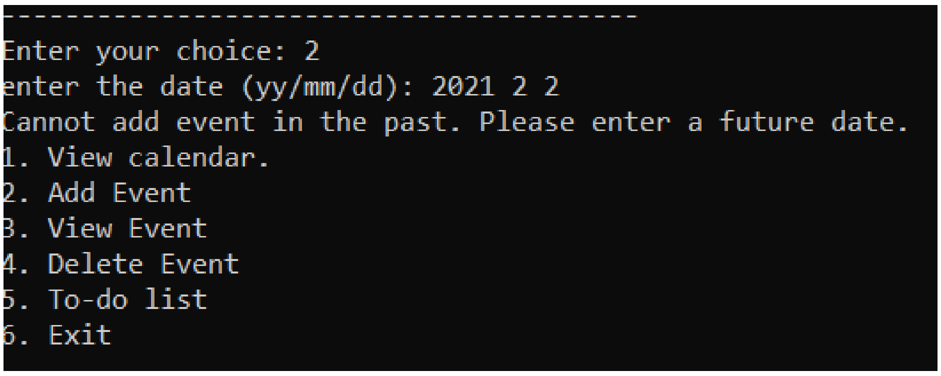


```
Enter year and month: 2025 7

----------------------------------------------
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
----------------------------------------------
```

Figure 2.4.1.B Calendar View

3.4.1.3 Date Validation

The system includes date validation to ensure that users cannot add events to past dates. It checks the entered date against the current system date to maintain logical consistency and prevent scheduling errors.

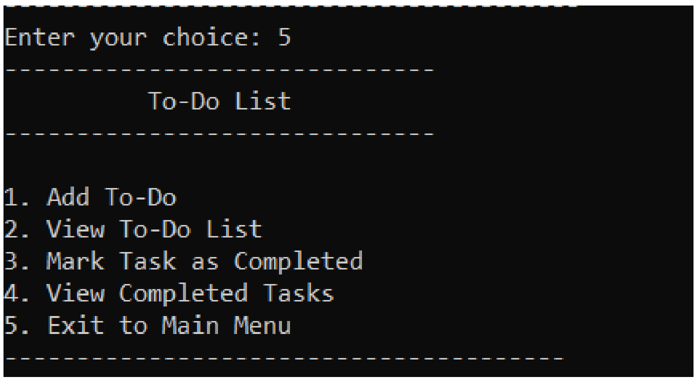


```
---------------------------------------
Enter your choice: 2
enter the date (yy/mm/dd): 2021 2 2
Cannot add event in the past. Please enter a future date.
1. View calendar.
2. Add Event
3. View Event
4. Delete Event
5. To-do list
6. Exit
```

Figure 3.4.1.C

3.4.1.4 To-Do List Menu

The to-do list menu provides options to add new tasks, view pending tasks, mark tasks as completed, and view completed tasks. It helps users manage daily activities efficiently within a simple and interactive interface.

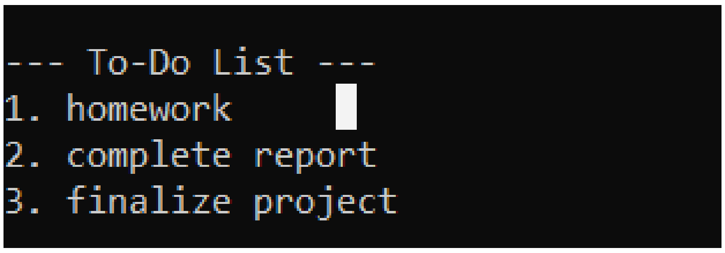


```
Enter your choice: 5
------------------------------
         To-Do List
------------------------------


1. Add To-Do
2. View To-Do List
3. Mark Task as Completed
4. View Completed Tasks
5. Exit to Main Menu
---------------------------------------
```

Figure 3.4.1.D To-do list menu

3.4.1.5 Viewing To-Do List

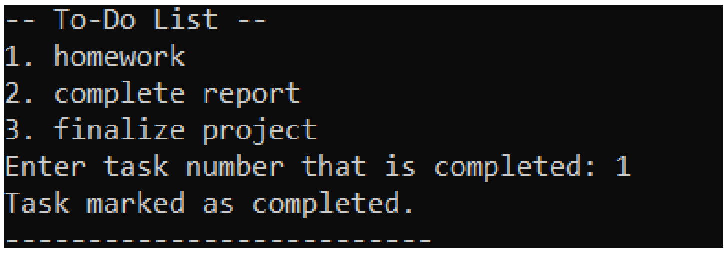
The viewing to-do list feature displays all pending tasks saved by the user. It reads data from the storage file and presents the tasks in a numbered list for easy tracking and management.



```
--- To-Do List ---
1. homework
2. complete report
3. finalize project
```

Figure 3.4.1.E View To-do list

3.4.1.6 Marking To-Do As Complete

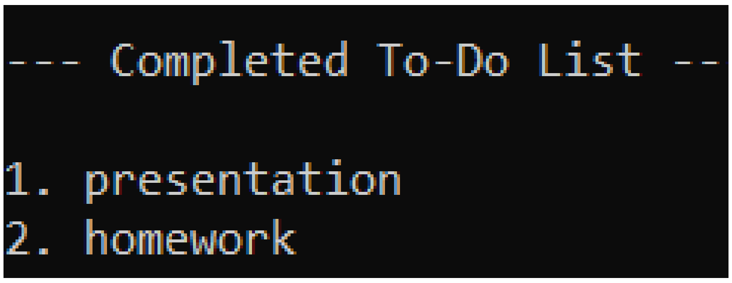


```
-- To-Do List --
1. homework
2. complete report
3. finalize project
Enter task number that is completed: 1
Task marked as completed.
--------------------------
```

Figure 3.4.1.F Mark To-do as complete

3.4.1.7 Completed To-Do

The completed to-do feature displays all tasks that have been marked as finished. It retrieves and lists these tasks from a separate file, allowing users to review their completed work and monitor productivity.



```
--- Completed To-Do List --

1. presentation
2. homework
```

Figure 3.4.1.G Completed TO-DO

3.4.2 Non-Functional Requirement

The Calendar Management System is designed not only to perform its intended functions but also to meet key non-functional requirements that enhance its overall quality, usability, and reliability:

- Usability: The system features a simple, text-based, menu-driven interface that is intuitive and easy to use, even for beginners. Clear prompts and logical navigation ensure a smooth user experience.
- Portability: Developed in standard C, the application runs seamlessly on both Windows and Linux platforms with minimal modifications. This ensures broad compatibility and ease of deployment.
- Maintainability: The application is built using modular code with separate functions for each feature (calendar, events, and to-do list). This makes it easy to identify, isolate, and fix issues or update features as needed.
- Reliability: Proper file handling, input validation, and error-checking mechanisms are in place to ensure the system behaves consistently and does not crash unexpectedly. Users are informed of issues through meaningful error messages.

- Performance: The system executes quickly and efficiently, utilizing basic C libraries without relying on any external dependencies. It is optimized for low memory usage and fast response times.
- Data Organization: Events and tasks are stored in structured text files, ensuring that data is logically organized and easy to retrieve. This design promotes clarity and consistency in data management.

CHAPTER 4: CONCLUSION

4.1 Problem Faced

- Managing file operations (read, write, and update) in plain text files without data loss.
- Correctly calculating weekdays and handling leap years for calendar display.
- Handling user input safely and avoiding input buffer issues with scanf and fgets.
- Implementing reliable event deletion without accidentally losing data.
- No built-in graphical interface, making UI limited to command-line formatting.
- Debugging logical errors and off-by-one mistakes took considerable time.
- Ensuring files exist before reading to avoid crashes.

4.2 Lesson Learned

- Importance of careful file handling to maintain data integrity in C.
- How to implement date and leap year calculations accurately.
- Managing user input effectively to avoid common pitfalls with scanf and fgets.
- The value of modular programming for organizing code logically.
- Necessity of validating user input and handling edge cases gracefully.
- How to debug and fix common logical errors like off-by-one mistakes.
- Limitations of command-line interfaces and the potential benefits of GUIs.
- Understanding platform-dependent functions and writing portable code.
- Patience and systematic testing are crucial for developing robust programs.

4.3 Conclusion

Working on this Calendar Management System project has been a great learning experience for us. It gave us the chance to explore different concepts in C programming, especially related to file handling, modular functions, and date management. We really enjoyed building something that’s practical and useful, even if it runs in a simple command-line interface.

With the support of our teachers and friends, we were able to complete the project successfully. It wasn’t just about writing code — we also learned how to structure a complete program, handle errors, and think about how users will interact with it.

These lessons will definitely help us in future academic projects and professional work.

At the same time, we understand that the project isn’t perfect. There are still many features we’d like to add — like reminders, a proper user interface, and maybe even online access. But as a starting point, this project helped us build confidence and gave us a strong foundation to work on more complex applications in the future.

4.4 Recommendation

- Implement a graphical user interface (GUI) for better user experience.
- Use a database (e.g., SQLite) instead of text files for more reliable data storage and querying.
- Add input validation to ensure dates and tasks are entered correctly.
- Include search and filter options for events and to-dos.
- Introduce recurring events and task priorities.
- Improve portability by avoiding platform-specific functions.
- Modularize code into separate files and use header files for better maintainability.
- Incorporate backup and restore features for event and to-do data.
- Add reminders or notification features for important dates and tasks.

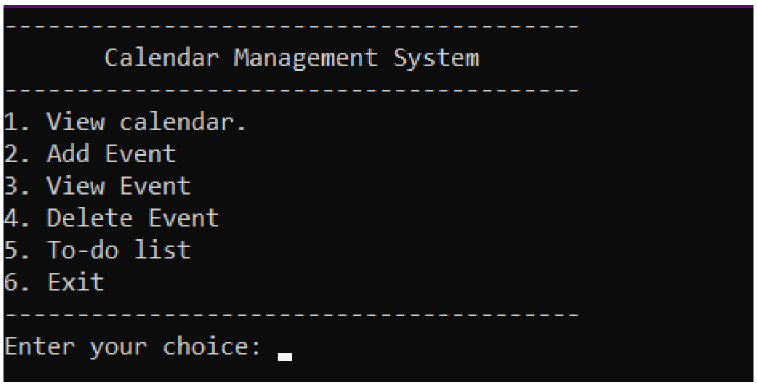
4.5 Future Enhancement

- Graphical User Interface (GUI): Develop a GUI version
- Calendar Navigation: Allow users to scroll through months and years interactively.
- Event Reminders: Add automatic reminders or notifications for upcoming events.
- Recurring Events: Support for daily, weekly, monthly, or yearly repeating events.
- Event Categories: Organize events by categories (e.g., personal, work, holidays).
- Priority-based To-Do List: Allow users to assign priorities or deadlines to tasks.
- Search Functionality: Enable searching for events or tasks by keyword or date.
- Cloud Sync/Backup: Store data online or back up files to prevent data loss.
- Mobile App Version: Create an Android or iOS version for on-the-go access.
- Voice Input Integration: Allow users to add tasks or events using voice commands.

APPENDIX

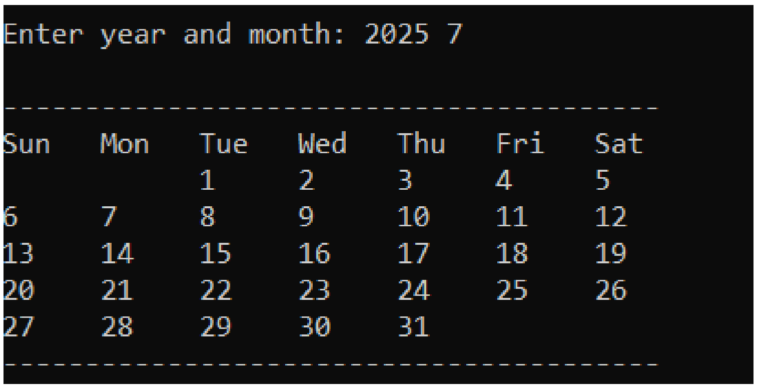
Snapshots of the System

Main menu



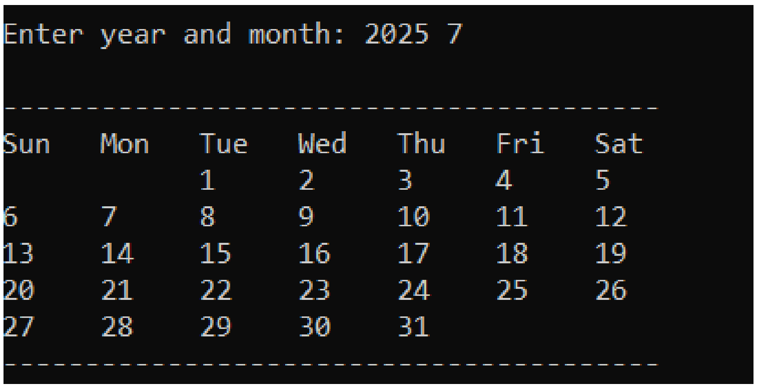
```
----------------------------------------
       Calendar Management System
----------------------------------------
1. View calendar.
2. Add Event
3. View Event
4. Delete Event
5. To-do list
6. Exit
----------------------------------------
Enter your choice:
```

Calendar view



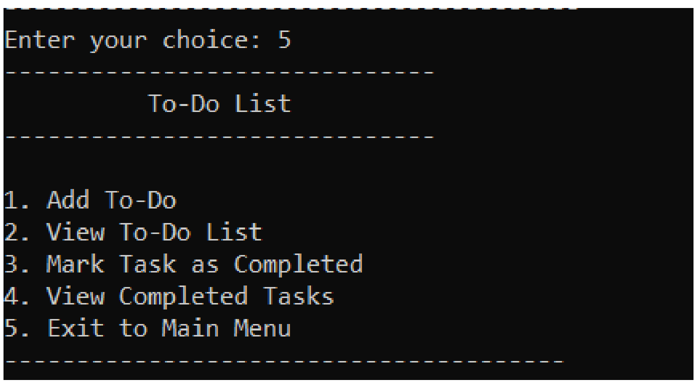
```
Enter year and month: 2025 7

----------------------------------------
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
----------------------------------------
```

Date Validation



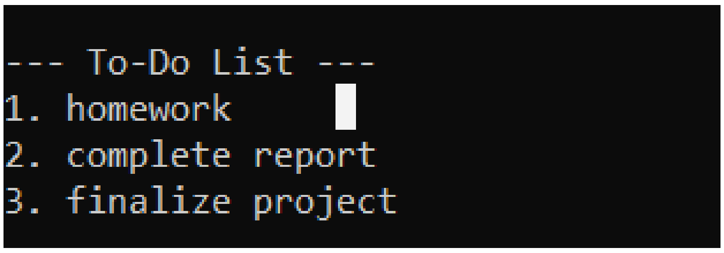
```
Enter year and month: 2025 7

----------------------------------------
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
----------------------------------------
```

To-Do List Menu



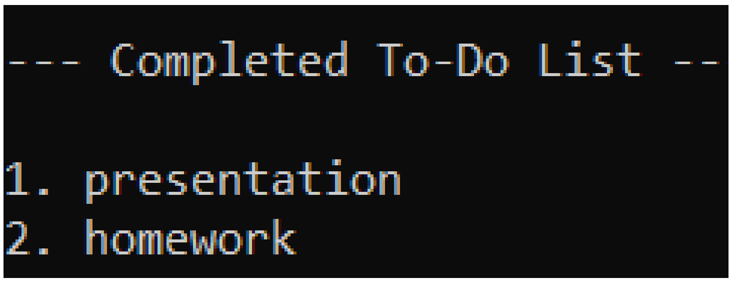
```
Enter your choice: 5
------------------------------
          To-Do List
------------------------------

1. Add To-Do
2. View To-Do List
3. Mark Task as Completed
4. View Completed Tasks
5. Exit to Main Menu
---------------------------------------
```

Marking To-Do as Completed



```
--- To-Do List ---
1. homework
2. complete report
3. finalize project
```

Completed To-Do



```
--- Completed To-Do List --

1. presentation
2. homework
```

BIBLIOGRAPHY

- For calendar.https://youtu.be/Zhecb3dMrao?si=F4EQQfgyThOvD7LM..
- Learning about new functions and syntax. Chatgpt.
- Book for reference – Let us c
- For time- https://www.google.com/search?client=opera&q=time+in+c&sourceid=opera&ie=UTF-8&oe=UTF-8
  Youtube- https://youtu.be/Qoed2uBwF_o?si=6cIjZmcANNjbeP0a

.