

Event Reminder System Documentation

Uppu Chaithanya Kumar

Lovely Professional University

Registration Number: 12113448

Training Organization: Cipher School

Summer Term 2025

Abstract

The Event Reminder System is a console-based C++ application developed during a summer training program at Cipher School. It enables users to manage events through adding, viewing, marking as done, deleting, and searching by title prefix using a trie data structure. This document details the system's design, implementation, and sample outputs, highlighting its use of advanced data structures for efficient event management.

Contents

1	Introduction	2
2	Features	2
3	Implementation	2
3.1	Event Class	2
3.2	Trie Class	2
3.3	ReminderManager Class	2
3.4	Main Function	2
4	Code Snippet	3
5	Sample Outputs	3
5.1	Adding an Event	3
5.2	Viewing Events	3
5.3	Marking an Event as Done	3
5.4	Viewing Completed Events	4
5.5	Searching by Prefix	4
6	Data Structures and Algorithms	4
7	Future Enhancements	4
8	Conclusion	5

1 Introduction

The Event Reminder System is a robust C++ application designed to help users organize and track events. Developed by Uppu Chaithanya Kumar (Lovely Professional University, Registration Number: 12113448) under the guidance of Cipher School, the system incorporates a trie for case-insensitive title searches and file persistence for data retention. This project showcases object-oriented programming and data structure concepts, making it a standout for the summer training program.

2 Features

- **Add Event:** Create events with title, description, and date (YYYY-MM-DD).
- **View Events:** Display all events with details and status.
- **Remove Event:** Delete an event by index.
- **Mark Event as Done:** Toggle completion status.
- **View Completed Events:** Show only completed events.
- **Search by Title Prefix:** Use a trie for case-insensitive prefix searches.
- **File Persistence:** Save and load events to/from `events.txt`.

3 Implementation

3.1 Event Class

Stores event details: title, description, date, and completion status. Methods include getters and a status toggle function.

3.2 Trie Class

Implements a trie for efficient, case-insensitive prefix-based title searches. The `toLowerCase` function ensures case-insensitive matching.

3.3 ReminderManager Class

Manages a vector of `Event` objects and a `Trie` for searches. Handles file I/O for persistence.

3.4 Main Function

Provides a menu-driven interface for user interaction, handling input validation and event operations.

4 Code Snippet

Below is a snippet of the `addEvent` function in `ReminderManager`:

```
1 void addEvent(Event e) {  
2     events.push_back(e);  
3     titleTrie.insert(e.getTitle(), events.size() - 1);  
4 }
```

5 Sample Outputs

The following outputs demonstrate the system's functionality:

5.1 Adding an Event

```
1 Welcome to Event Reminder Management System  
2 1. Add Event  
3 2. View Events  
4 3. Remove Event  
5 4. Mark Event as Done  
6 5. View Completed Events  
7 6. Search Events by Title Prefix  
8 7. Exit  
9 Enter your choice: 1  
10 Enter title: Birthday chaitu 13 sept  
11 Enter description: chaithanya's birthday 13 september  
12 Enter date (YYYY-MM-DD): 2025-09-13
```

5.2 Viewing Events

```
1 Enter your choice: 2  
2 Event 1:  
3 Title: Birthday chaitu 13 sept  
4 Description: chaithanya's birthday 13 september  
5 Date: 2025-09-13  
6 Status: Pending  
7  
8 Event 2:  
9 Title: gangamma jatara  
10 Description: gangammathalli jatara.  
11 Date: 2025-05-15  
12 Status: Pending
```

5.3 Marking an Event as Done

```
1 Enter your choice: 4  
2 Event 1:
```

```

3 | Title: Birthday chaitu 13 sept
4 | Description: chaithanya's birthday 13 september
5 | Date: 2025-09-13
6 | Status: Pending
7 |
8 | Event 2:
9 | Title: gangamma jatara
10 | Description: gangammathalli jatara.
11 | Date: 2025-05-15
12 | Status: Pending
13 |
14 | Enter the event number to mark as done: 2

```

5.4 Viewing Completed Events

```

1 | Enter your choice: 5
2 | Event 2:
3 | Title: gangamma jatara
4 | Description: gangammathalli jatara.
5 | Date: 2025-05-15
6 | Status: Completed

```

5.5 Searching by Prefix

```

1 | Enter your choice: 6
2 | Enter title prefix to search: birth
3 | Event 1:
4 | Title: Birthday chaitu 13 sept
5 | Description: chaithanya's birthday 13 september
6 | Date: 2025-09-13
7 | Status: Pending

```

6 Data Structures and Algorithms

- **Trie:** Enables case-insensitive prefix searches with $O(p + n)$ time complexity, where p is the prefix length and n is the number of matches.
- **Vector:** Stores events for $O(1)$ access and $O(n)$ iteration.
- **File I/O:** Uses `fstream` for persistent storage with `|` as a delimiter.

7 Future Enhancements

- Integrate a Qt-based GUI for a user-friendly interface.
- Add a background thread for real-time reminders.

- Sort events by date using a map.
- Implement date format validation with regex.

8 Conclusion

The Event Reminder System fulfills the requirements of the Cipher School summer training program while incorporating advanced features like a trie-based search and file persistence. It demonstrates proficiency in C++ and data structures.