Malnad College of Engineering

(An Autonomous Institution under Visvesvaraya Technological University, Belagavi)

Hassan, Karnataka, India – 573202

Course Title: Data Structures

Course Code: 23AI304

Project Based learning

"Title:Implementation of Listing the event name,venue
And timings of JAMBOREE 2024 and allow students to register
For the events."

Submitted by:

M Sadhamail	4MC23CI026
K B Shreya	4MC23CI021
Samrudhi K S	4MC23CI047
Sinchana B	4MC23CI052

Submitted to:

Dr. Balaji Prabhu B V

Associate professor and HOD

Dept. of CSE (AI and ML)

Department of Computer Science and Engineering

(Artificial Intelligence and Machine Learning)

Malnad College of Engineering

PB# 21, Hassan, Karnataka, India – 573 202

2024-2025

Introduction

Problem statement :

The schedule of Jamboree 2024 is give in a text file. Implement a program with appropriate data structures to demonstrate the following functionalities:

1. List all events: Parse the file and display the event names.

2. Get venue and timings of a specific event: Allow users to input the event name and retrieve its venue and timings. Handle cases where the event name is invalid.

3. Register students for an event: Maintain a dictionary or another data structure to track registrations. Allow users to add student names to specific events.

4. Generate total registrations for each event: Display a summary of the total registrations for all events.

Dataset used Implement the above task:

Jamboree Event schedule 2024:

20/12/2024,Maneesh Mela,Main block,5:00-7:00 PM.

21/12/2024,Art Attack,Cricket ground,1:00-2:00 PM.

21/12/2024,Rangoli,Eco park,2:00-3:00 PM.

21/12/2024,You have got a minute,Civil Seminar hall,2:00-3:00 PM.

21/12/2024,Quiz,AB classroom,2:00-3:00 PM.

21/12/2024,E Parodgy,Civil Seminar hall,3:00-3:30 PM.

21/12/2024,Telli Flix,AB classroom,3:00-3:30 PM.

21/12/2024,Lock and Roll,Main stage,4:30-5:30 PM.

21/12/2024,Bolly via,Civil Seminar hall,4:30-5:30 PM.

21/12/2024,Conquest,AB classroom,4:30-5:30 PM.

21/12/2024,Glamour in the shadows,Main stage,5:30-6:45 PM.

21/12/2024,Thala mela,Main stage,6:45-8:00 PM.

Source :

The Event list of Jamboree 2024 is obtained from their insta page and copied into a textfile to implement the given task.

Data Structures

Array of Structures:

The program uses an **array of structures** to store Events name and their venue and timings of the events in memory. Each structure contains 6 fields:

- **Date:** Stores the date when the event is conducted.
- **Event:** Stores the event name.
- **Venue:** Stores the venue of the event.
- **Time:** Stores the time when the event is going to take place
- **Student name:** Stores the name of the students who register for the particular event.
- **Count:** Stores how many students registered for the particular event.

Definination Of Sturcture:

```
struct Jam
{ char date[100];
  char event[100];
  char venue[100];
  char timings[100];
  char student_name[50];
   int st_count;
} Events[500];
```

Why array of structure is used?

- An **array** stores multiple elements of the same type, while a **structure** can store elements of different types. An **array of structures** allows us to store multiple records, each containing different data types, in a contiguous block of memory.

- In this program ,we have a set of Events , each event have date, name,venue and timings, which are different types of data (character, integer). An array of structures allows us to group all this data together for each event.

Steps involved in the program

1.Program Overview

The program is a basic event registration system with the following functionality:

- **List all events**: Displays all available events.
- **Get the venue and timings of the events**: Shows the venue and timings of a specific event when searched by the event name.
- **Student Registration to the events**: Allows students to register for an event by providing their name.
- **Display the summary of the registrations**: Displays a summary of how many students have registered for each event.

2. Struct Definition (`struct Jam`)

- **Purpose**: This structure holds information about an event (date, event name, venue, timings) and student registration details (`student_name`, `st_count` which tracks the number of students registered).
- **Array of Structures**: `Events[500]` is an array of `struct Jam`, designed to hold information for up to 500 events.

3. Global Variable (`line`)

- **Purpose**: This variable is used to track the number of lines read from the `Events.txt` file, which presumably contains event information. It helps in determining how many events have been loaded.

4. Main Function

- **Menu system**: The program uses a simple text-based menu with 5 options. The user selects an option to perform various tasks (view events, get details, register, or display summary).
- **Infinite loop**: A `for(;;)` loop is used to keep the menu running until the user selects "Exit". The `switch` statement handles different menu options.

5. Listing Events (`list_events`)

- **File Handling**: This function opens a file `Events.txt` that presumably contains a list of events. It reads the file line-by-line using `fgets` and prints each line to the console.
- **Line Counting**: The `line` variable is incremented as each event is read. This count will be used later for processing event registrations.

6. Getting Event Details (`get_details`)

- **File Handling**: This function reads from `fullevents.txt`, which contains detailed information about events (e.g., date, event name, venue, timings). It parses each event into the `Events` array.
- **Event Search**: The user is prompted to enter the name of the event they want details about. The program searches the `Events` array for a matching event and prints its details.

7. Student Registration (`register_stud`)

- **Student Registration**: The function first prompts the user for the event name and the student's name.
- **Event Search**: It searches for the event in the `Events` array using `strstr` to match the event name. When a match is found, the student's name is stored in the `student_name` field of the corresponding `struct Jam`. The student count (`st_count`) is then incremented.

8. Displaying Registration Summary (`display`)

- **Display Registered Students**: This function prints the registration summary for each event, including the event name and the number of students who have registered (`st_count`).

Pseudo code

1.START

2.Defining a structure:

Define a structure "Jam" to store event details:

- Attributes: date, event name, venue, timings, student name, student count.

Initialize an array of "Jam" structures to hold events.

- Events[500]

- line = 0 (to track the number of lines/events)

Structure Definition (struct Jam)

- The Jam structure stores details of each event:
  - date: The date of the event.
  - event: The name of the event.
  - venue: Where the event will take place.
  - timings: When the event will happen.
  - student_name: Name of the student registered for the event.
  - st_count: Number of students registered for the event.
- We have defined an array Events[500] to store details for up to 500 events.
- Events[500]: Array to store all event details.
- line: Keeps track of the number of events read from the file or registered.

3.Predefining the Functions:

.Define the following functions: - list_events()

get_details()

register_stud()

display()

4.In the Main Function:

4. Main function:

LOOP forever:

a. Display menu: 1. List all events 2. Get details (venue and timings) of a specific event 3. Register a student for an event 4. Display summary of registrations 5. Exit the program

b. Input user's choice.

c. Use SWITCH-CASE to call the appropriate function based on the choice.

d. Exit if the choice is 5.

In Main Function

- **Purpose**: Provides a menu to interact with the program.
- **Menu Options**:
    1. List all events (`list_events`)
    2. Get venue and timings of a specific event (`get_details`)
    3. Register a student for an event (`register_stud`)
    4. Display the summary of registrations (`display`)
    5. Exit the program.
- **Logic**:
    - The program enters an infinite loop to continuously prompt the user for a choice.
    - Based on the user's choice, it calls the respective function.
    - If the user chooses option 1 ,all the events in Jamboree 2024 will be listed.
    - If the user chooses option 2,The name of a particular event will be taken as input and if found the particular events details i.e date,venue and timings will be displayed.
    - If the user chooses option 3,Student can register for an event.
    - If the user chooses option 4,Summary of events and total number of registration will be displayed .
    - If the user chooses option 5, the program exits.

5.List function( ):

list_events():

- Open the file containing event names.

- Read the file line by line.

- Print each line (event name).

- Increment the line count.

- **Purpose**: Displays a list of events from a file (`Events.txt`).
- **How It Works**:
  - Opens the file `Events.txt` in read mode.
  - Reads each line from the file using `fgets` and prints it to the screen.
  - Increments the global `line` variable to keep track of the number of events read.

6.Getdetails function( ):

get_details():

- Open the file containing full event details. - Input the event name to search.

- Parse the file to populate the "Events" structure array.

- Search the array for the given event.

- IF found: Print the event details (date, venue, timings).

- ELSE: Print "Event not found."

- **Purpose**: Provides detailed information (date, venue, timings) about a specific event.
- **How It Works**:
  1. Opens the file `fullevents.txt` which contains detailed event data.
  2. Reads event details (date, name, venue, timings) line by line and stores them in the `Events` array.
  3. Prompts the user to input the name of the event they want details for.
  4. Searches the `Events` array for a match using `strstr` (checks if the input event name is part of any stored event).
  5. If found:
     - Prints the details (date, venue, timings).
  6. If not found:
     - Prints "Event not found."

7.Register students function( ):

register_stud():

- Input event name and student name.

- Search the "Events" array for the given event name.

- IF found: Add the student name to the event and increment the student count.

- ELSE: Print "Event not found."

- **Purpose**: Registers a student for a specific event.
- **How It Works**:

1. Prompts the user to input the event name and their name.
2. Searches the `Events` array for the input event using `strstr`.
3. If the event is found:
    - Copies the student's name into the `student_name` field of the corresponding `Events` entry.
    - Increments the `st_count` (student count) for that event.
4. If not found:
    - Prints "Event not found."

8.Display function():

display():

- Print the registration details for all events.

- For each event in the "Events" array:

Print the event name and the number of students registered.

- **Purpose**: Displays a summary of registrations for all events.
- **How It Works**:

- Iterates over the `Events` array.
- For each event, it prints:
    - The event name.
    - The number of students registered (`st_count`).

9.END.

OUTPUT

->Main function():

```
PS C:\Users\HP\Desktop> gcc e.c
PS C:\Users\HP\Desktop> ./a out
1.List all events
2.Get the venue and timings of the events
3.Student Registeration to the events
4.Display the summary of the registerations
5.Exit the program
Enter your choice
1
```

Displays the menu for the user to know which choice to enter.

->Listing the events:

```
1.List all events
2.Get the venue and timings of the events
3.Student Registeration to the events
4.Display the summary of the registerations
5.Exit the program
Enter your choice
1
Jamboree Event schedule 2024:
Maneesh Mela
Art Attack
Rangoli
You have got a minute
Quiz
E Parodgy
Telli Flix
Lock and Roll
Bolly via
Conquest
Glamour in the shadows
Thala mela
```

For choice 1.All the events of Jamboree 2024 are listed

->Get details():

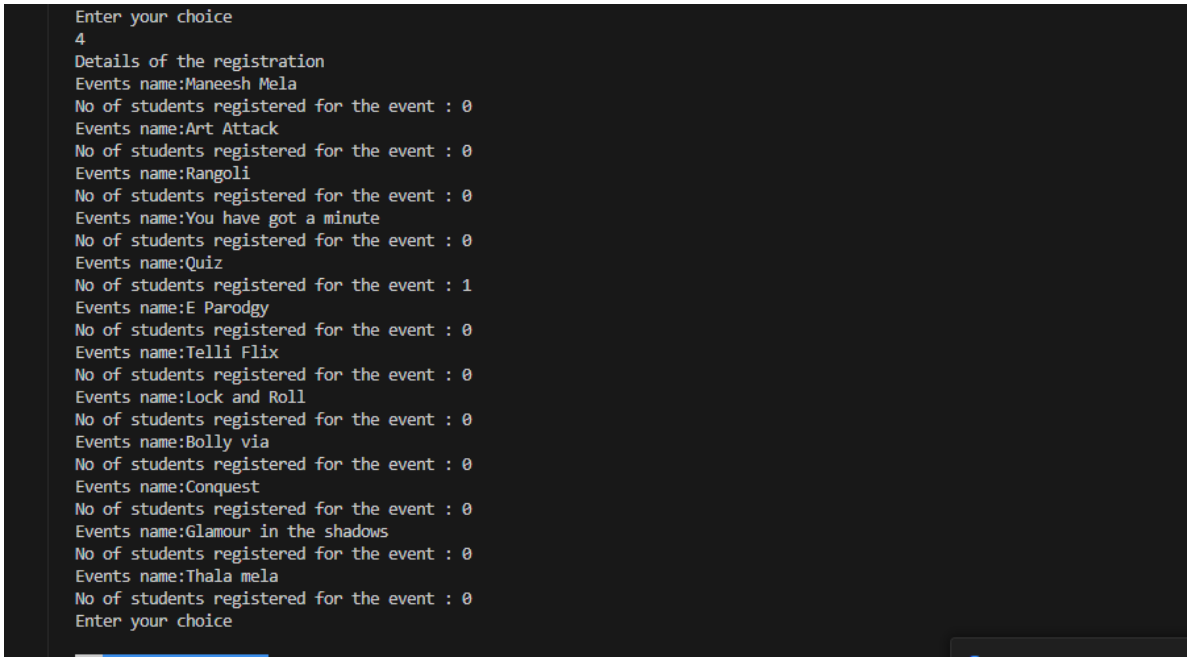
```
Enter your choice
2
Enter the event
Quiz
Event found!
Event date: 21/12/2024
Event Name: Quiz
Venue: AB classroom
Timings: 2:00-3:00 PM.
Enter your choice
```

For choice 2 all the details of QUIZ event is displayed.

->Regs_student():

For choice 3,Student ammu is registered under the event Quiz

->Display():



For choice 4,details of events and number of students registered for the event will be printed.

->To Exit:

When the user enter any choice other than choices 1-4 the program execution exits.

Conclusion:

This program provides a simple implementation for managing events and student registrations. It demonstrates basic file handling, structure usage, and user input handling in C. However, for practical usage, it would benefit from enhancements such as dynamic memory allocation, better error handling, and persistent storage of updated data.

This event management program serves as a basic yet effective tool for handling event-related operations in an organized manner. It simplifies the process of event tracking, student registrations, and summary generation, making it particularly useful in educational institutions, clubs, or small-scale organizations.

- **Streamlined Event Management:**
    - By centralizing event data, the program reduces manual efforts in maintaining and searching for event details. Users can list all available events or query specific event details like venue and timing with ease.
- **Student Engagement:**
    - The student registration functionality fosters active participation in events by providing a straightforward way to sign up. This encourages students to stay involved and engaged in institutional activities.
- **Data Organization:**
    - Event details are organized and stored in a structured format, ensuring clarity and easy access. This improves overall efficiency in managing multiple events simultaneously.
- **Insights into Participation:**
    - The summary feature provides valuable insights into the level of interest and participation for each event. This information can be useful for event organizers to assess the popularity of events and plan future activities accordingly.

And some more benefits like this…