Event Planner & Budget Tracker

Authors:

1. Maulshree Sharma 2401020218
2. Keshav Maini 2401020233
3. Pratyaksh Tyagi 2401020232
4. Sarthak Sabharwal 2401020234

Affiliation: JIIT Noida
Batch: A-8

Semester: 2nd
Date: 30th April 2025

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY
jiit
विद्या तत्व ज्योतिसम:

Table of Contents

Problem Statement

Event management often requires coordinating various components such as attendee tracking, venue selection, service management, and budgeting. Traditionally, event organizers rely on spreadsheets or manual records, leading to inefficiencies, inconsistencies, and data loss. Additionally, staying within budget while selecting suitable venues and services can be challenging without a proper system. Therefore, a digital solution that automates these tasks and ensures budget control, data persistence, and ease of access is essential.

Objective

The primary objective of this project is to develop a C++ application that serves as an event planner and budget tracking tool. The system should allow users to:

- Create and update events with all relevant details.
- Manage lists of attendees with extended attributes like guest details and security needs.
- Select services and venues based on predefined availability and pricing within a user-specified budget.
- Persist data using file handling for event, venue, and service information.
- Provide a user-friendly, menu-driven interface with clear data presentation.

Features of C++ used

Feature	Description
Object-Oriented Programming	Use of multiple classes (Event, Attendee, Venue, etc.) for modular design.
Encapsulation & Abstraction	Data and functions are encapsulated in classes, exposing only necessary interfaces.
Inheritance & Modularity	Structured and extendable class definitions make future scaling easier.
File Handling	Persistent storage using .csv files through fstream operations.
Formatted Output	Tables and columns displayed using iomanip (setw, left, etc.).
Conditional Statements	Used for login validation, menu control, budget logic.
Loops	For iterative input (e.g., adding attendees) and navigation.
Exception-like Handling	Login attempts are limited with feedback messages and early exit logic.

Code Workflow

Startup

- Launches with main() function.
- Calls login() for user authentication.

Login

- Maximum of 3 attempts.
- If authenticated, proceeds to welcome().

Welcome

- Displays a brief welcome message and project intro.
- Proceeds to main_menu() after user input.

Main Menu

- Prompts user with 3 choices:
    1. Create a new event → newEvent()
    2. Manage existing event → existingEvent()
    3. Exit

New Event Creation

- Calls enterEventDetails() to gather:
    - Event name, host, contact, date, time, location, type, and budget.
- Then calls addAttendeeList() to populate attendees.
- Calls chooseVenueAndServices() to select options within budget.
- Data is saved to files:
    - event.csv for event info.
    - <event_name>_attendee.csv for attendee info.
    - v_and_s_<host_initial>_<date>.csv for selected services and venue.

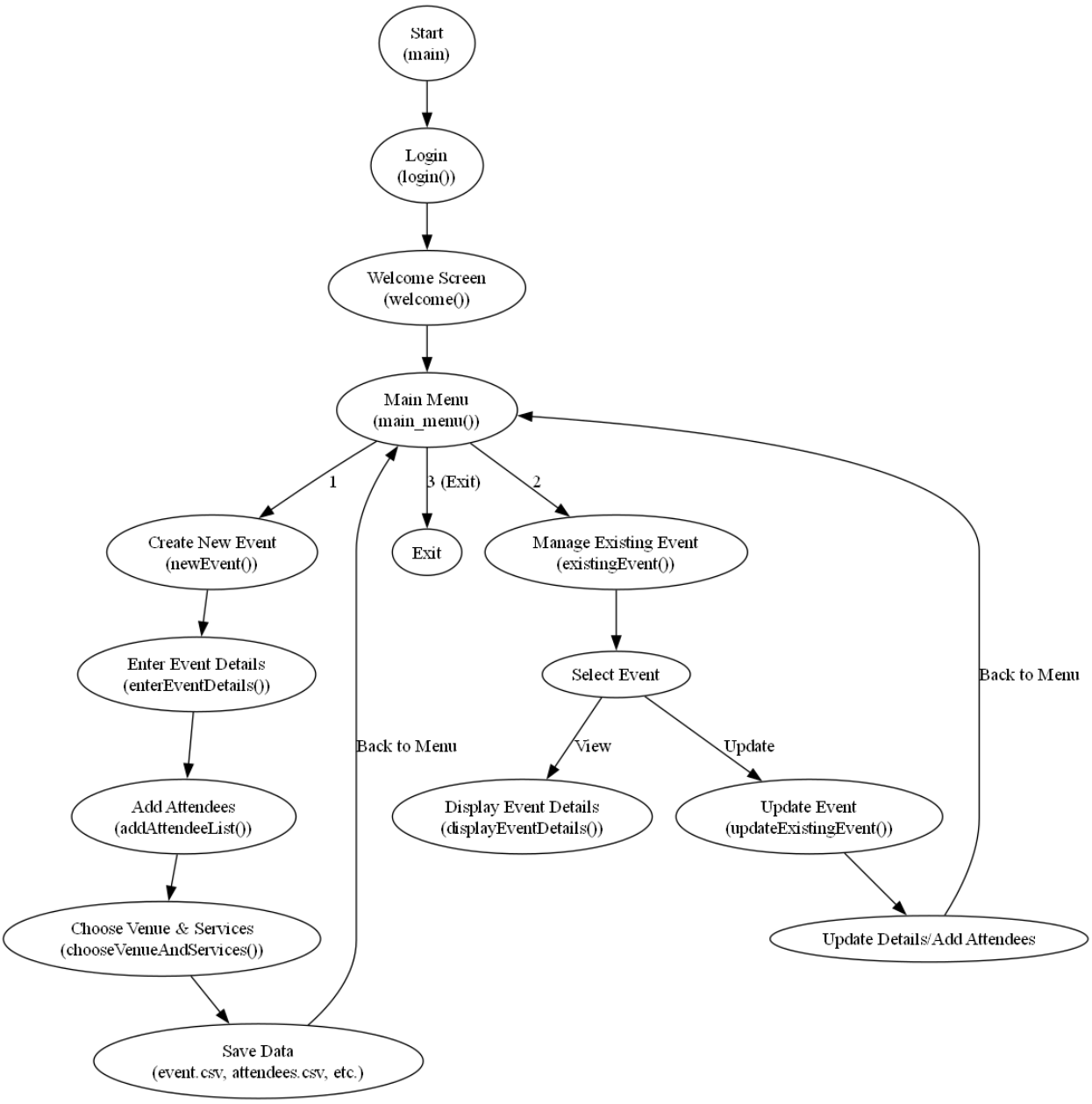
Existing Event Management

- User selects event from a list.
- Options:
    - View details → displayEventDetails()
    - Update event → updateExistingEvent(), which calls:
        - updateEventDetails() → edit core event info.
        - addAttendeeList() → modify attendee list.

Exit

- Gracefully terminates the program.

Program Flowchart



Classes & Their Responsibilities

Attendee

- Attributes: name, contact, rsvp, hasGuest, guestName, needsSecurity
- Methods:
    - display(): Prints formatted attendee information.

Services

- Attributes: name, contact, type, price, availability, rating
- Methods:
    - display(): Shows service details.

Venue

- Attributes: name, location, contact, capacity, price, availability, rating
- Methods:
    - display(): Shows venue details.

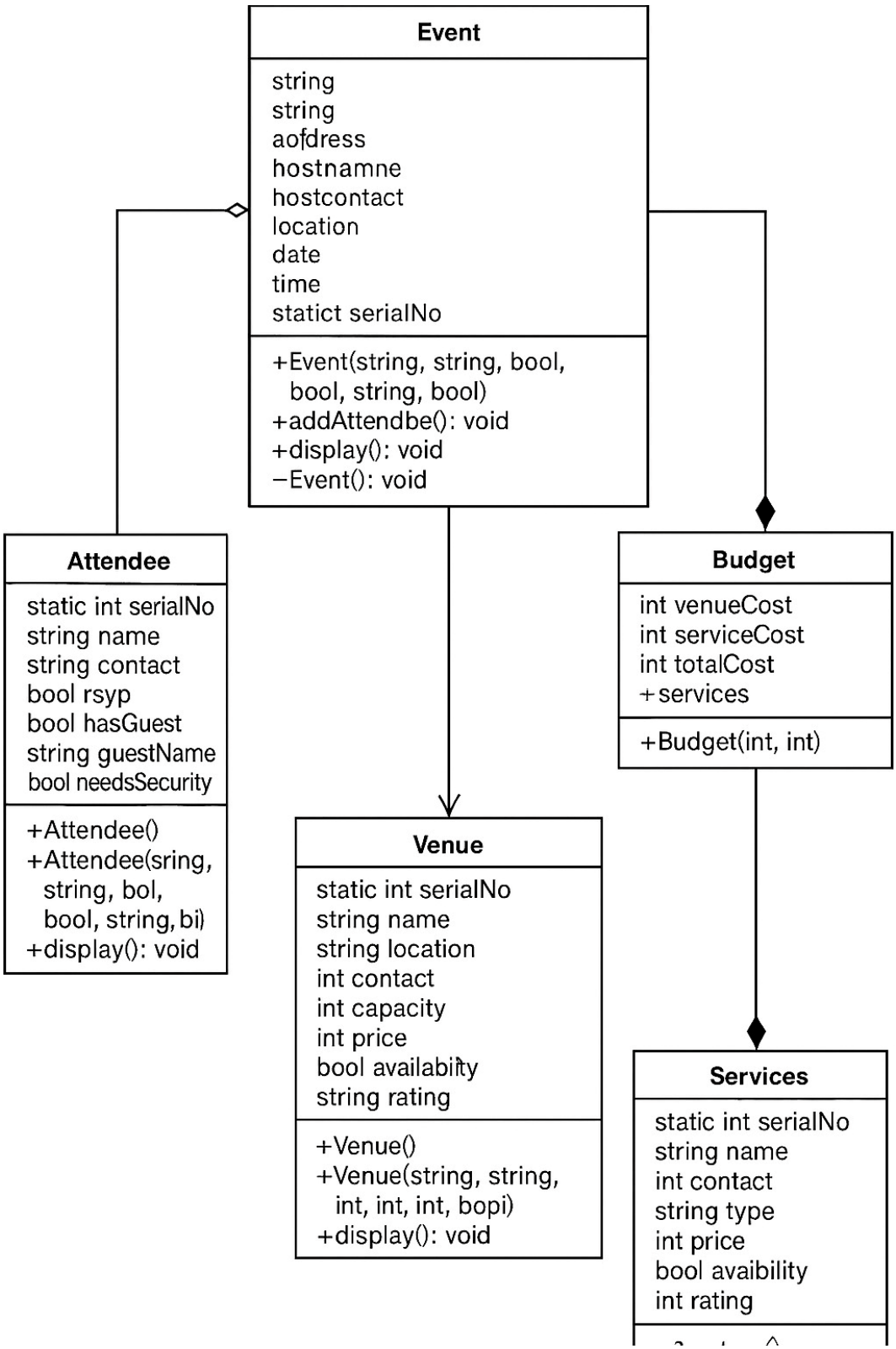
Budget

- Attributes: venueCost, serviceCost, totalCost, services[]
- Methods:
    - calculateTotal(): Computes total with 10% commission.
    - applyDiscount(): Reduces 5% if total exceeds ₹10,00,000.

Event

- Attributes: name, hostname, hostcontact, location, date, time, type, attendees[], budget, venue
- Methods:
    - addAttendee(): Adds attendees.
    - display(): Prints full event summary.

Class Diagram

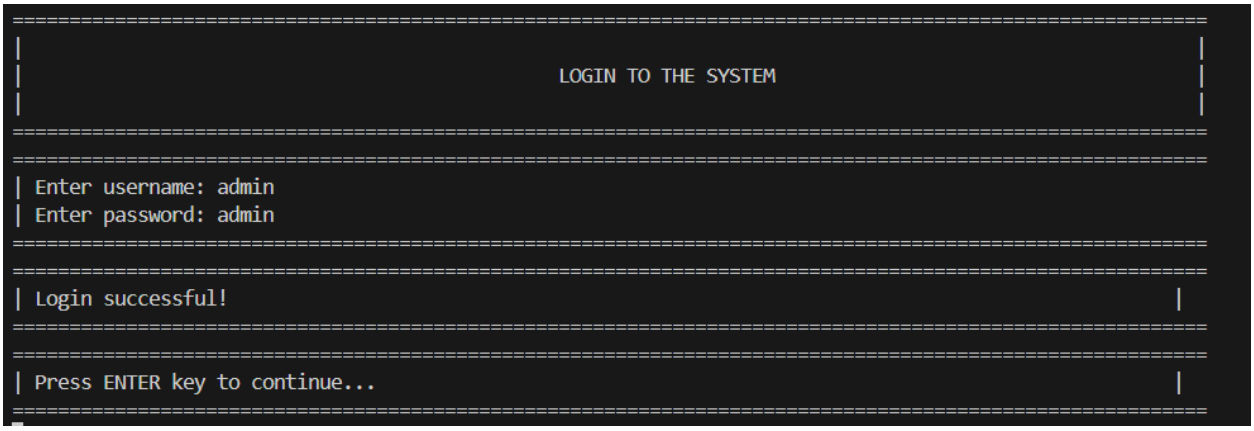


List of Functions & Purpose

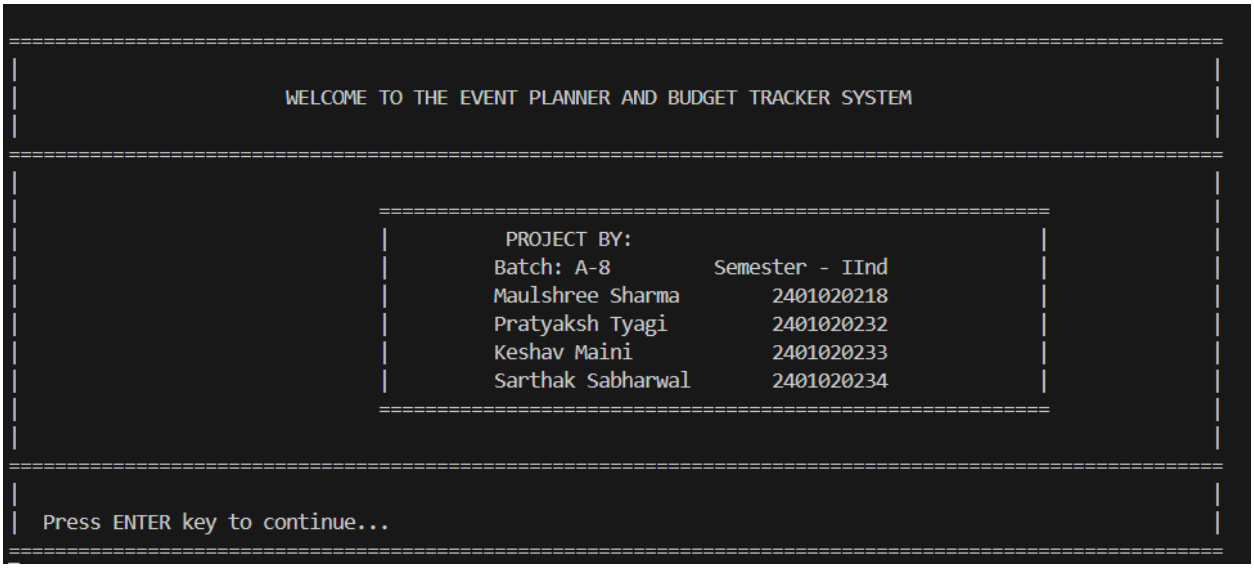
Function Name	Purpose
login()	Authenticates the user (max 3 attempts).
welcome()	Displays welcome message.
main_menu()	Displays options and routes control.
newEvent()	Main driver for creating a new event.
enterEventDetails()	Takes event details from user and initializes the event object.
addAttendeeList()	Adds multiple attendees for the current event.
chooseVenueAndServices()	Selects venues and services within budget constraints.
existingEvent()	Displays and allows editing of previously created events.
displayEventDetails()	Reads and displays event data from file.
updateExistingEvent()	Allows user to select and update an event.
updateEventDetails()	Edits core event fields (name, location, etc.).

Sample Output Screens

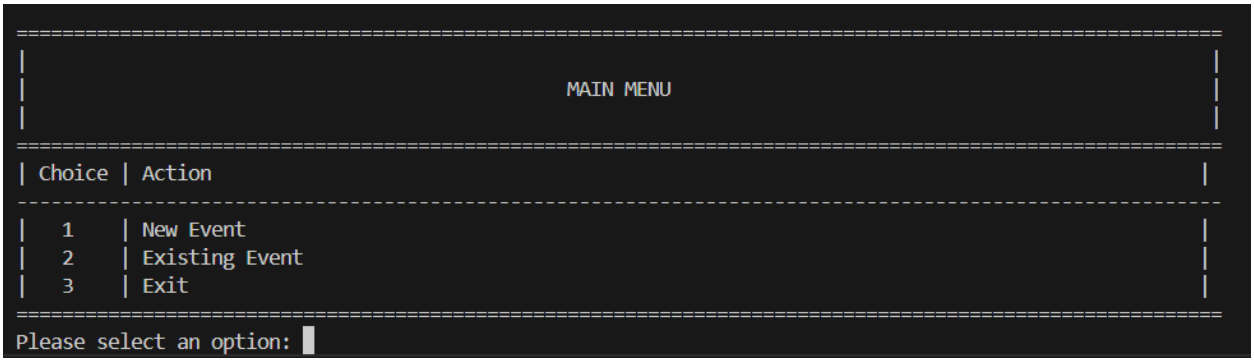
Login Screen



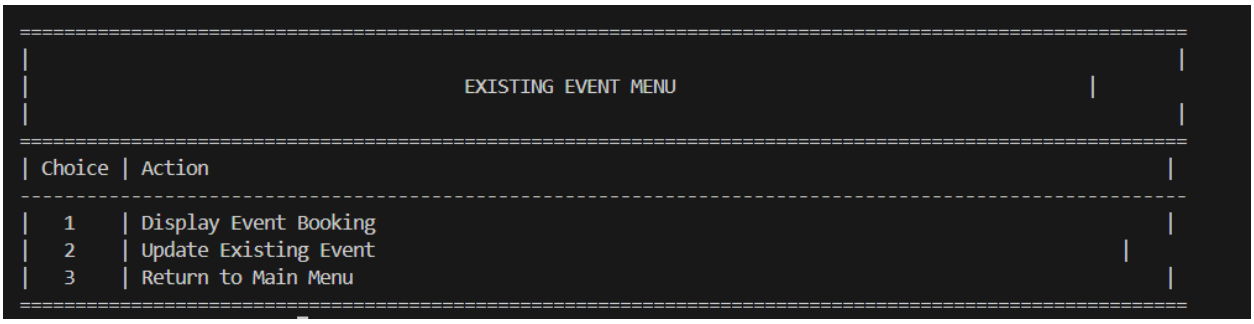
Welcome Screen



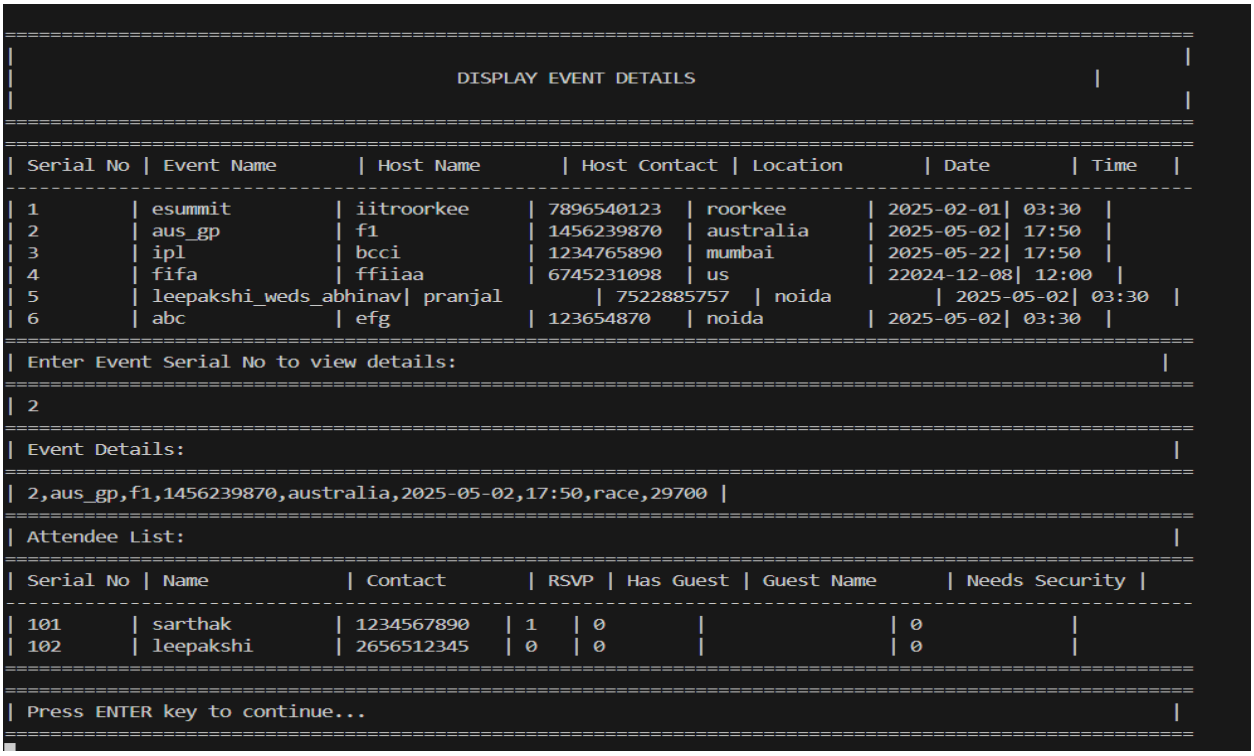
Main Menu



Existing Event Menu



Display Event Details



Conclusion

The Event Planner and Budget Tracker System in C++ is a powerful tool that integrates multiple aspects of event management. With an object-oriented structure, effective file handling, and a clear UI, the system demonstrates the practical application of C++ in developing real-world software. The modularity ensures easy maintenance and scope for future enhancements such as:

- Integration with a GUI library (e.g., Qt).
- Use of databases instead of CSV for large-scale use.
- Web-based extension with backend logic in C++.