STUDENT EVENT PARTICIPATION DASHBOARD

A Mini Project Report

Submitted by

KARTHIKEYAN S

KEERTHNA S

in association with Database Management System

IN
ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

RAJALAKSHMI ENGINEERING COLLEGE
[Autonomous]

Regulations 2023

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this report title "**STUDENT EVENT PARTICIPATION DASHBOARD**" is the Bonafide work of **KARTHIKEYAN S** (**2116231801081**) and **KEERTHNA S** (**2116231801083**) who carried out the mini project work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. J M GNANASEKAR ,
HEAD OF THE DEPARTMENT,
Professor,
Department of AI&DS,
Rajalakshmi Engineering College
Chennai – 602 105.

SIGNATURE

Mrs.M.THAMIZHARASI,
SUPERVISOR,
Assistant Professor,
Department of AI&DS,
Rajalakshmi Engineering College,
Chennai – 602 105.

Submitted for the DBMS Mini project review held on

Internal Examiner **External Examiner**

ACKNOWLEDGEMENT

Initially I thank the Almighty for being with us through every walk of my life and showering his blessings through the endeavor to put forth this report.

My sincere thanks to our Chairman **Mr. S. MEGANATHAN, M.E., F.I.E.,** and our Chairperson **Dr. (Mrs.)THANGAM MEGANATHAN, M.E., Ph.D.,** for providing me with the requisite infrastructure and sincere endeavoring educating me in their premier institution.

My sincere thanks to **Dr.S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our work in time.

I express my sincere thanks to **Dr. J M Gnanasekar M.E.,Ph.D.,** Head of the Department of Artificial Intelligence and Data Science for his guidance and encouragement throughout the project work. I convey my sincere and deepest gratitude to our internal guide, **Mrs.M.Thamizharasi.,M.Tech., (Ph.d).,.,**Assistant Professor, Department of Artificial Intelligence and Data Science, Rajalakshmi Engineering College for his valuable guidance throughout the course of the project.

Finally I express my gratitude to my parents and classmates for their moral support and valuable suggestions during the course of the project.

ABSTRACT

In academic institutions, tracking student engagement in events and activities is crucial for fostering an active learning environment and enhancing student development. However, managing student records manually is labor-intensive and often inaccurate, leading to inefficiencies in assessing student participation and making data-driven decisions. The **Student Event Management Dashboard (SEMD)** was developed as a solution to streamline the management of student event data, enabling efficient record-keeping, filtering, and visualization.

The primary motivation for this project stems from the need to automate the storage and retrieval of student participation records, which can be challenging to maintain in large educational institutions where students are involved in numerous events throughout the academic year. Manually tracking this data not only requires significant time and resources but also lacks the scalability to handle growing volumes of information. In conclusion, the **Student Event Management Dashboard** fulfils a critical need in modern educational institutions by providing an automated solution for managing and analysing student event participation. SEMD not only reduces the administrative burden associated with manual record-keeping but also equips institutions with insights into student engagement, facilitating strategic decision-making.

TABLE OF CONTENTS

S NO.	TITLE
1	Abstract
2	Introduction
3	Project Outline
4	Implementation
5	Overview of languages used
6	Analysis
7	Database Implementation
8	Output Analysis and Explanation
9	Conclusion

INTRODUCTION

Traditionally, student participation records have been managed manually or through rudimentary digital means, such as spreadsheets. While these methods may suffice for smaller institutions, they quickly become impractical and error-prone as the number of students and events increases. In large educational institutions, the manual tracking of event participation data often results in delays, inaccuracies, and an increased workload for administrative staff. Moreover, the lack of a streamlined, automated system limits the institution's ability to analyse participation trends and make data-driven decisions, such as identifying popular event types or understanding engagement levels across different student demographics..

The **Student Event Management Dashboard (SEMD)** includes real-time data visualization features, which dynamically display participation trends through plots, helping administrators gain valuable insights into student engagement.

This project leverages a combination of web development and database management technologies, including HTML, CSS, JavaScript, Python, and a relational database management system (RDBMS). The three-tier architecture adopted for SDMS – comprising the frontend, backend, and database layers – ensures that the system is modular, scalable, and efficient.

Through the development and deployment of the **Student Event Management Dashboard**, this project aims to not only simplify the management of student records but also empower educational institutions to utilize participation data in meaningful ways. By automating the process of data entry, filtering, and visualization, SEMD enables administrators to focus on strategic planning and student engagement initiatives, rather than on manual data management. This report discusses the design, implementation, and testing of the system, demonstrating its utility as a powerful tool for modern educational institutions seeking to optimize their student data management practices.

PROJECT OUTLINE

The **Student Event Management Dashboard (SEMD)** project is designed to provide an efficient, web-based solution for managing and analysing student participation in academic events. The system aims to automate the traditionally manual process of record-keeping, reducing administrative workload and minimizing errors. SEMD features a user-friendly interface where administrators can input, retrieve, and filter data based on roll numbers, event participation modes (online or offline), and date ranges. This filtered data is displayed through real-time visualizations, allowing institutions to gain valuable insights into student engagement trends.

The architecture of SEMD is built on a three-tier structure comprising a frontend, backend, and database. The frontend, developed using HTML, CSS, and JavaScript, provides a responsive and intuitive interface for data input and display. The backend, implemented in Python, handles data processing, applies filters, and generates visualizations using data from a relational database. The database layer supports structured data storage and efficient retrieval, ensuring reliable and organized record management.

The system leverages technologies like Python for server-side operations and SQL for database interactions, creating a robust and scalable solution. The SEMD demonstrates the potential of integrating web development and database management to facilitate effective student data management and enable data-driven decision-making in educational institutions.

IMPLEMENTATION

1. Requirement Analysis

- Identified the need for an automated system to manage and visualize student participation data.
- Gathered user requirements, such as data input for roll numbers, event types, and date filters.
- Defined the scope of functionalities, including data filtering, visualization, and basic user interaction.

2. Planning

- Outlined the three-tier architecture: frontend, backend, and database.
- Chose appropriate technologies: HTML, CSS, JavaScript for the frontend; Python for backend logic; SQL for database management.
- Established a timeline for developing each component, ensuring integration and testing phases were included.

3. Database Design

- Created a relational database schema with tables for student records, events, and participation data.
- Defined primary and foreign keys for data integrity and structured relationships between tables.
- Developed SQL queries for data insertion, retrieval, and filtering based on user inputs.

3. Backend Development

- Utilized Python (with frameworks such as Flask) to build the server-side application.
- Implemented endpoints to receive form submissions, process filters, and query the database.
- Integrated data visualization libraries (e.g., Matplotlib) to generate participation plots dynamically.

5. Frontend Development

- Designed the web interface using HTML and CSS for a responsive and intuitive layout.
- Added JavaScript to handle form submissions and facilitate asynchronous communication with the backend.
- Ensured that the user interface allowed for easy input of roll numbers, event modes, and date ranges.

6. User Authentication (Optional/Advanced)

- Implemented a basic user authentication system to ensure secure access.
- Used session management in Python to track user logins and permissions.
- Planned for role-based access control to limit functionalities to authorized users only.

OVERVIEW OF LANGUAGES USED

1. HTML (Hypertext Markup Language)

- **File**: index.html
- **Role**: HTML is used to create the structure and layout of the web page for the **Student Event Management Dashboard.** It serves as the foundation of the frontend, defining the form fields where users can input roll numbers, select participation modes (online or offline), and specify date ranges for filtering participation data.
- **Functionality**: HTML elements in index.html include:
    - Input fields for roll numbers, mode selection (e.g., radio buttons or dropdown menus), and date pickers for selecting date ranges.
    - A submission button that triggers the form submission to the backend. ○ Basic page structure, such as headers, paragraphs, and containers, to organize the content for readability.

2. CSS (Cascading Style Sheets)

- **File**: Embedded within index.html or as a separate linked file (not provided here but assumed as part of frontend styling).
- **Role**: CSS is used to style the HTML elements on the web page, making the interface visually appealing and responsive. It controls the appearance, including colors, fonts, spacing, and layout, ensuring that the form is user-friendly and accessible on different devices.
- **Functionality**: CSS enables customization of:
    ○ The look and feel of the input fields, buttons, and overall page design. ○ Responsive behavior to adjust the page layout for different screen sizes (desktop, tablet, mobile). ○ Hover effects or transitions for interactive elements to improve user experience.

3. JavaScript

- **File**: JavaScript is often embedded in index.html or included as an external file.
- **Role**: JavaScript is used to add interactivity to the webpage, enabling client-side form validation, data collection, and sending data to the backend.
- **Functionality**: JavaScript is responsible for:
    - Collecting data from form fields and preparing it to be sent to the backend.
    - Sending an HTTP request to the backend (app.py) when the user submits the form, allowing asynchronous communication between the client and server.
    - Optionally, displaying feedback to users (like error messages if the input is incorrect or a confirmation message on submission).

4. Python

- **File**: app.py
- **Role**: Python is used to build the backend of the **Student Event Management Dashboard .** It handles server-side logic, processes data, applies filters based on the user's input, and generates data visualizations.
- **Functionality**:
    - **Web Framework**: Typically, a lightweight Python web framework such as Flask is used in app.py to handle HTTP requests from the frontend.
    - **Data Processing**: Python processes input data from the user, retrieves relevant records from the database, and applies filters (e.g., by roll number, event mode, and date range).
    - **Data Visualization**: Python can use libraries like Matplotlib or Plotly to generate graphs or plots that visually represent student participation trends. These visualizations are then returned to the frontend as images or embedded plots.
    - **Response Handling**: After processing the data, app.py sends back a response to the frontend, which could include filtered data or an updated participation plot.

5. SQL (Structured Query Language)

- **File**: SQL statements are typically embedded in the Python backend (app.py) to interact with the database.

- **Role**: SQL is used to communicate with the relational database where student participation records are stored. It facilitates data storage, retrieval, and filtering, supporting efficient database operations.
- **Functionality**:
    - **Data Storage and Retrieval**: SQL queries are used to store new records into the database and retrieve records based on criteria specified by the user.
    - **Data Filtering**: SQL helps filter data by roll number, event mode, and date range as per user inputs.
    - **Data Management**: It enables efficient data management by organizing records into tables with a structured schema, ensuring data integrity and reliability.

Summary of Languages and Their Roles

Language	File	Purpose
HTML	index.html	Defines the structure of the webpage and user input form.
CSS	index.html (or external)	Styles the HTML content, providing a user-friendly design.
JavaScript	index.html	Adds interactivity, collects user inputs, and sends requests to the backend.
Python	app.py	Handles backend logic, processes data, interacts with the database, and generates visualizations.
SQL	Embedded in app.py	Manages database operations such as data storage, retrieval, and filtering based on user criteria.

ANALYSIS

1. System Architecture

The architecture of SEMD consists of three main layers:

- **Frontend**: Built with HTML, CSS, and JavaScript, the frontend provides an interface for users to interact with the system. It allows users to input data, such as roll numbers, event modes (online or offline), and date ranges, to filter and retrieve student participation records. JavaScript adds interactivity, managing data submission and displaying feedback to users.
- **Backend**: The backend, developed in Python, processes incoming requests from the frontend. It handles data retrieval from the database, applies user-defined filters, and generates visualizations. The backend also manages communication with the frontend, ensuring data is correctly sent and received.
- **Database**: The database stores all student participation records, using SQL for data storage, retrieval, and filtering. This structured relational database supports efficient data management and enables complex queries to retrieve data based on multiple parameters.

2. Functionality and Features

The SEMD incorporates key functionalities that simplify data management and enhance user experience:

- **Data Filtering**: Users can filter data by specific criteria, including roll number, event mode, and date range. This feature allows administrators to retrieve relevant records quickly, enabling them to analyze specific groups of students or types of events.
- **Real-Time Data Visualization**: The system provides data visualization, displaying participation trends as charts or graphs. This feature helps users identify patterns, such as attendance trends over time or participation by event type, facilitating data-driven decision-making.
- **User-Friendly Interface**: The web interface, designed with HTML and CSS, offers a clean, intuitive layout, making it easy for users to input data and access information. The responsive design ensures that the system functions smoothly across various devices, including desktop and mobile platforms.

3. Strengths of the System

The SEMD project has several strengths that enhance its effectiveness:

- **Automation**: By automating the record-keeping process, the system reduces manual data entry and retrieval time, allowing administrators to focus on other important tasks.
- **Efficiency and Speed**: The backend efficiently processes user requests, filtering data based on multiple criteria and returning results promptly. This efficiency improves the user experience and ensures quick access to information.
- **Scalability**: The system is designed to handle an expanding volume of student data, making it scalable as the institution grows. The use of a relational database allows for organized, structured data storage, which can support future enhancements.
- **Data Visualization**: Visual representations of participation trends provide valuable insights into student engagement, helping educators understand attendance patterns and adjust event planning accordingly.

4. Limitations of the System

Despite its strengths, the SEMD has some limitations:

- **Limited Filtering Options**: While the system allows filtering by roll number, event mode, and date range, additional filters (such as event categories or class levels) could provide more granularity in data analysis.
- **Static Data Visualization**: The current visualization setup may offer limited interactivity. For example, administrators may benefit from dynamic visualizations where they can adjust parameters directly on the chart or zoom in on specific sections of data.
- **Single-User Focus**: As it stands, SEMD might not support multiple concurrent users effectively, depending on the deployment method. This limitation could impact performance if the system is accessed by a large number of users at once.

- **Security Concerns**: The project might lack robust security measures, such as authentication and access control, which are critical for protecting sensitive student information.

5. Potential Improvements

To address its limitations and enhance functionality, the following improvements could be considered for future iterations of the system:

- **Enhanced Filtering Options**: Adding more filters, such as event types, student departments, or academic years, would make the system more versatile and allow for deeper analysis.
- **Interactive Data Visualization**: Incorporating interactive charts using libraries such as Plotly or D3.js would allow users to interact directly with the data visualization. This enhancement could include features like zooming, hovering over data points, and filtering directly from the graph.
- **Multi-User Support and Access Control**: Implementing user authentication and role-based access control would improve security and allow multiple users (e.g., faculty, administrators, students) to access the system with appropriate permissions.
- **Performance Optimization**: Optimizing SQL queries and backend logic could further enhance the speed and responsiveness of the system, especially when handling large datasets. Caching frequently accessed data might also improve performance.
- **Enhanced Security**: Adding security features, such as input validation, encryption, and secure data storage, would ensure the protection of student data and prevent unauthorized access.

DATABASE IMPLEMENTATION

The **Student Event Management Dashboard (SEMD)** relies on a robust database to manage and store event participation records effectively. SQL (Structured Query Language) was used to create and populate the student table, which holds essential data about student participation in workshops, webinars, seminars, conferences, and other academic activities. This section outlines the SQL queries utilized to insert comprehensive data into the database, ensuring accurate and structured storage for efficient retrieval and analysis.

SQL DATA INSERTION QUERIES:

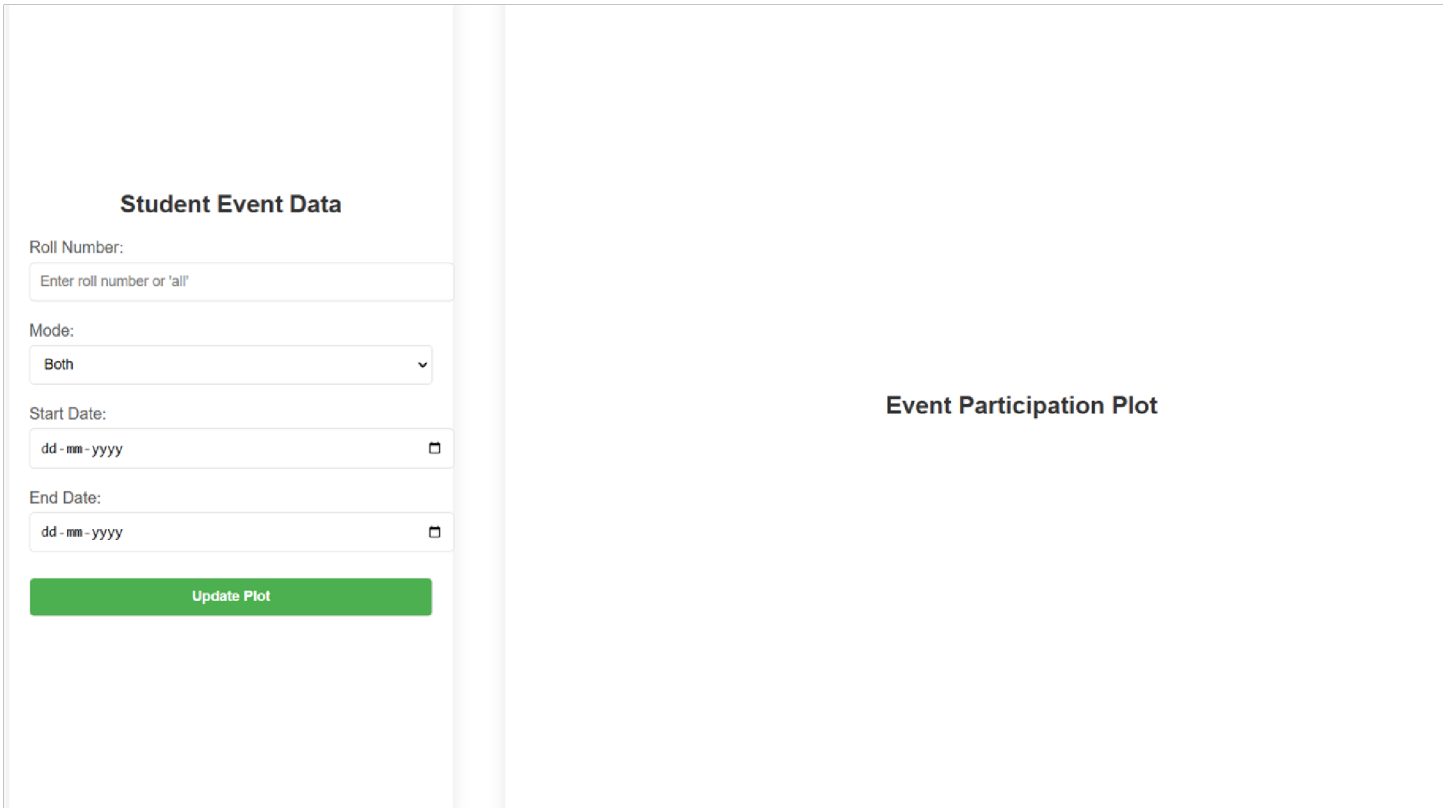
```
USE Student_details;

INSERT INTO student (si_no, roll_no, name_of_student, dept, type_of_event, title_of_event, from_date, to_date, mode, organization, certificate_upload)
VALUES
(1, 'S001', 'Alice Johnson', 'CSE', 'Workshop', 'AI and Machine Learning Workshop', '2023-01-05', '2023-01-07', 'Online', 'Tech University', 'Yes'),

(2, 'S001', 'Alice Johnson', 'CSE', 'Webinar', 'Deep Learning Techniques', '2023-02-12', '2023-02-12', 'Online', 'Data Science Institute', 'No'),

(3, 'S002', 'Bob Smith', 'ECE', 'Conference', 'International Robotics Conference', '2023-03-01', '2023-03-03', 'Offline', 'RoboTech Corp', 'Yes'),

(4, 'S002', 'Bob Smith', 'ECE', 'Workshop', 'Embedded Systems Design', '2023-04-10', '2023-04-12', 'Online', 'Tech Innovators', 'Yes'),

-- (Include other entries up to the last record)

(50, 'S010', 'Jane Miller', 'ME', 'Workshop', 'Future of Manufacturing', '2025-02-15', '2025-02-17', 'Online', 'ManufactureTech', 'Yes');
```

OUTPUT ANALYSIS AND EXPLANATION

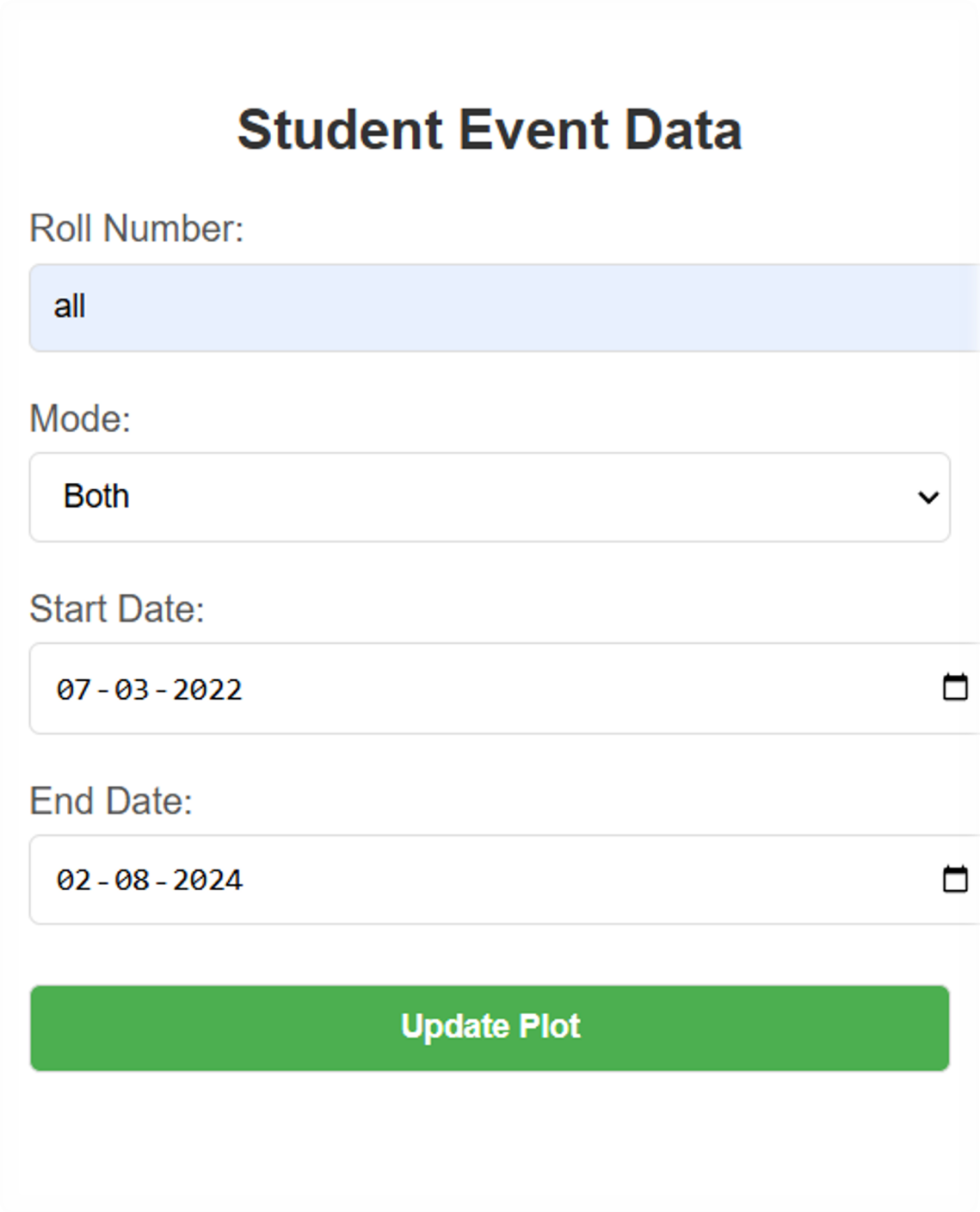
After successfully executing the HTML and Python scripts in conjunction with the database, the **Student Event Management Dashboard (SEMD)** generates outputs that reflect student participation data. The outputs obtained include:

1. **WEB INTERFACE OVERVIEW**

The web interface allows users to input roll numbers, select participation modes (online or offline), and specify date ranges to filter student event records. The following outputs are displayed:



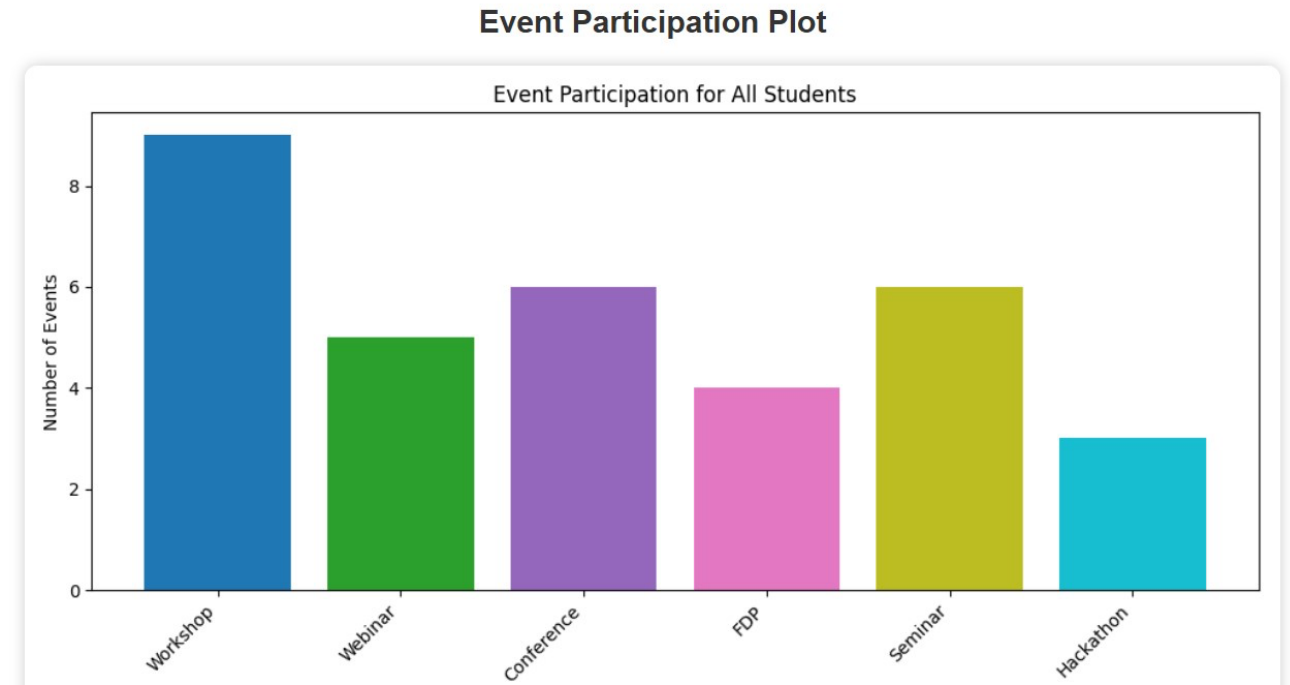
2. USER INTERFACE



3. EVENT PARTICIPATION PLOT

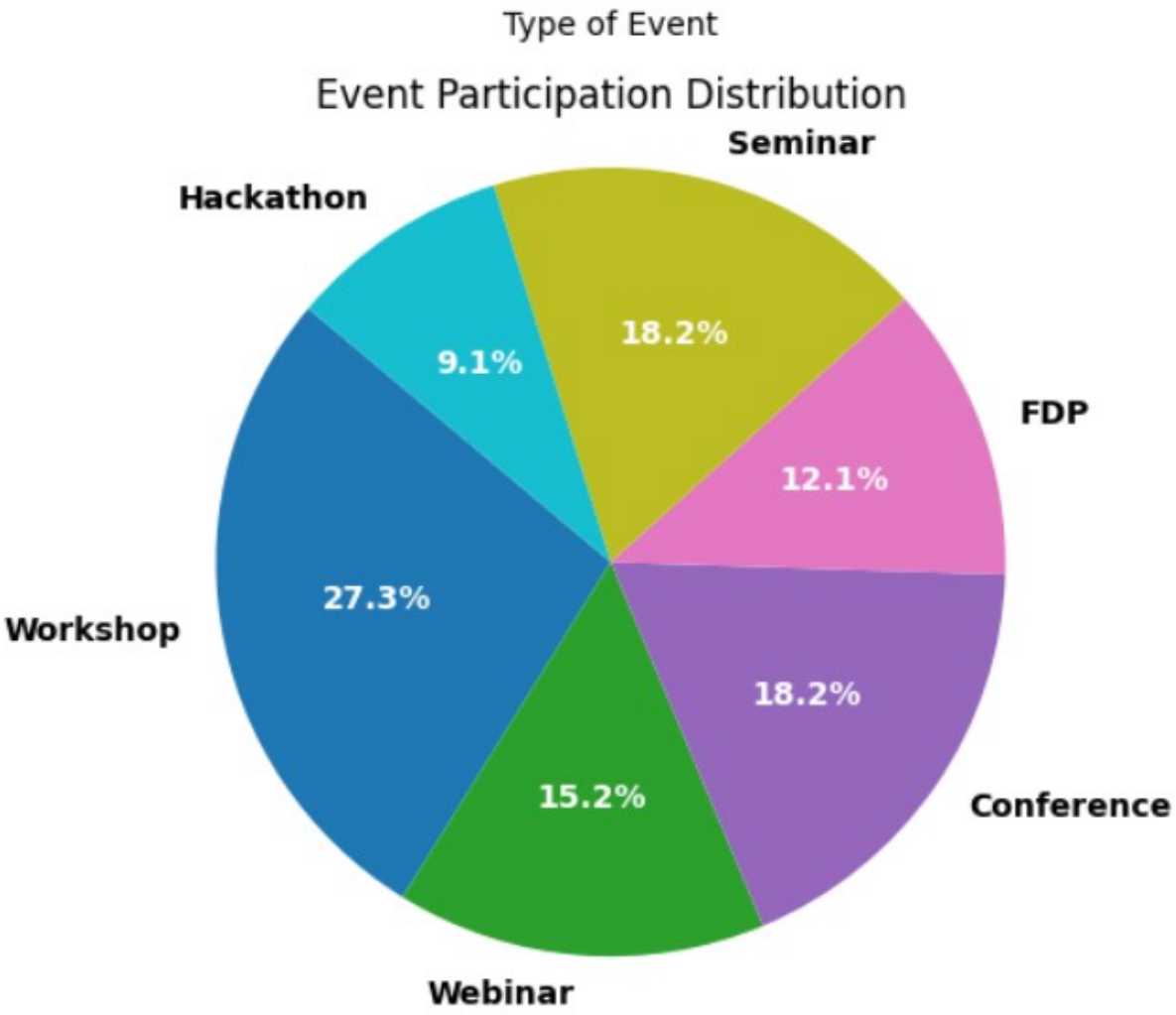
The event participation plot provides a graphical representation of student engagement across selected events. This visualization is generated using Python's data visualization libraries (e.g., Matplotlib) and includes:

- **X-Axis**: Represents the event types.
- **Y-Axis**: Represents the number of participants or event frequency.
- **Legends**: Indicate event modes (e.g., online, offline).



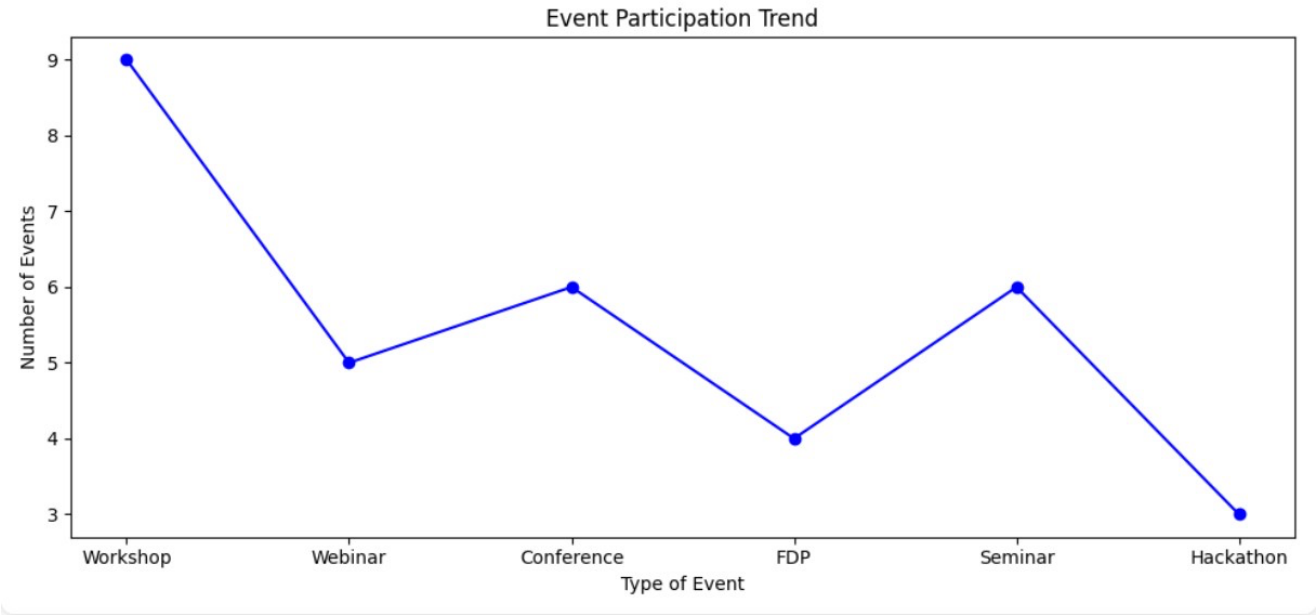
4.EVENT PARTICIPATION DISTRIBUTION

The spread and frequency of student involvement across various events, showcasing how participation is divided among different event types, departments, or time periods. This analysis helps in understanding which events attract the most attendees and reveals patterns that can guide future event planning and is implemented in a pie chart:



5.EVENT PARTICIPATION TREND

Line and bar charts effectively illustrate the number of students participating in events over a specific timeline, providing a visual representation of engagement trends



CONCLUSION

The **Student Event Management Dashboard (SEMD)** provides an effective solution for managing and analyzing student participation data in academic settings. By automating the data entry, filtering, and visualization processes, the system significantly reduces the time and effort required for administrative tasks. Through a user-friendly interface, administrators can easily filter records by roll number, event mode, and date range, enabling targeted analysis of student engagement across different events. Additionally, the real-time data visualization component presents clear insights into participation trends, which can support datadriven decisions in event planning and student outreach.

The SEMD's architecture, which combines a web-based frontend, Python backend, and a relational database, offers a scalable foundation capable of handling a growing volume of data as institutional needs evolve. This system highlights the value of integrating database management and web technologies to streamline record-keeping and improve data accessibility in educational environments. While the SEMD achieves its primary objectives, there are opportunities for future enhancements, such as adding more granular filters, incorporating interactive data visualization, and implementing multi-user support with secure access control. These improvements would make SEMD an even more robust and versatile tool for academic institutions, enhancing its ability to manage student data comprehensively.

In summary, the **Student Event Management Dashboard** effectively addresses the challenge of tracking student participation, delivering a reliable, scalable, and insightful solution that can grow alongside institutional requirements.