Aptech
Unleash your potential

TECHVVIZ 6
GLOBAL AI-BASED TECH COMPETITION

Software Requirements Specification

Version 1.0

EventSphere

Theme: College Event Information System
Category: Full-Stack Application Development

Table of Contents

1.1 Background and Necessity for the Full-Stack Web Application

Colleges and universities frequently organize technical competitions, cultural fests, academic events, and more. However, details about such events are often shared through noticeboards or group messages, leading to missed updates and low participation. Additionally, when the staff has to manage these events manually, it often leads to miscommunication, resource mismanagement, scheduling conflicts, and poor student engagement.

For students who are keen to participate in events, searching for accurate information and schedules can also be cumbersome in case of manually managed systems.

There may also be students who just want to view information about past and upcoming events without any intention of participating. In a manually managed system, they find it difficult to locate the information they are seeking.

Few issues and challenges with a traditional/manually managed system are as follows:

- **Ineffective Communication Channels**: Event details are often circulated through noticeboards, word-of-mouth, or scattered group messages leading to missed announcements and confusion among students.

- **Low Participation Rates**: Lack of timely and consistent updates results in low student turnout and missed opportunities for those genuinely interested in participating.

- **Manual Event Management Challenges**:
  - Staff face difficulties coordinating schedules, managing resources, and keeping track of registrations.
  - High potential for miscommunication and scheduling conflicts.
  - Limited visibility into past events or participation analytics.

- **Limited Accessibility for Students**: Students eager to participate must rely on informal sources or manually seek out information from faculty or peers.

- **Passive Participants are Overlooked**: Students who simply wish to browse past or upcoming events (for awareness or interest) find it difficult to locate such information in a manual system.

To overcome all these challenges, a centralized College Event Information System is necessary which provides real-time access to event information.

The system will serve as a digital platform where students, faculty, and other authorities can easily access event details, schedules, register for events, and receive notifications, all in one place.

1.2 Proposed Solution

The proposed solution is to develop a fully functional Web-based College Event Information System titled '**EventSphere'**. It will display upcoming and past events, allow student registrations for events, and enable administrators to add/edit/delete events. It will also provide features such as event categorization, notifications, image galleries, and so on. The Web app will be a comprehensive platform to obtain information regarding events, manage college events, and also aid interested participants in the due process.

1.3 Purpose of the Document

The purpose of this document is to outline the requirements for development of the **EventSphere** Full-stack application. This document will serve as a guide for the development team, ensuring that all stakeholders have a clear understanding of the project's objectives and functionalities.

This document explains the purpose and features of **EventSphere**, its interfaces, what the application will do, and the constraints under which it must operate. This document is intended for both stakeholders and developers of the application.

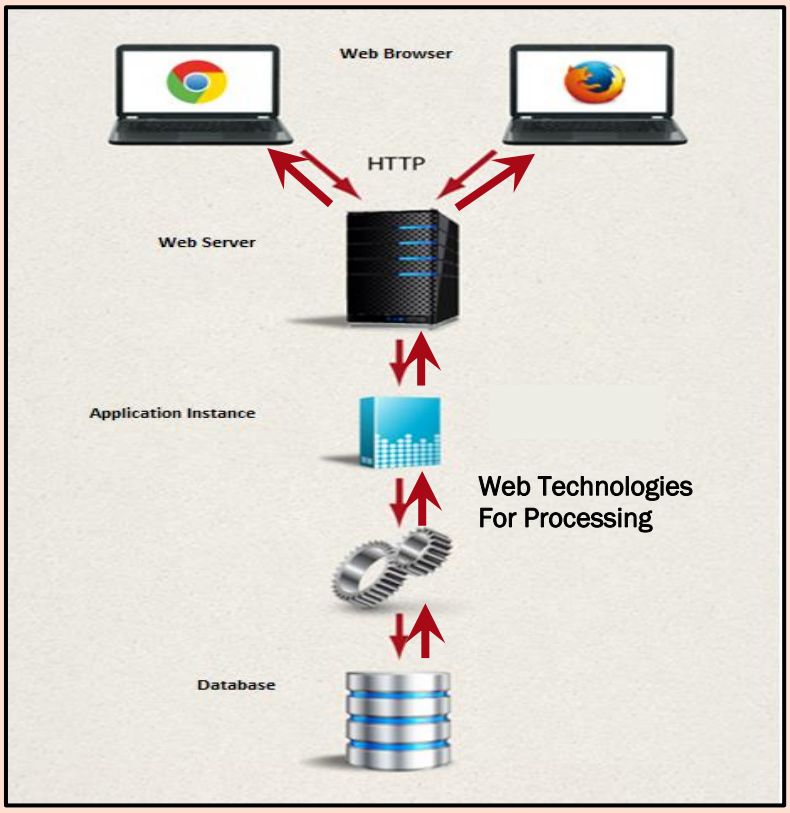
1.4 Scope of the Project

EventSphere will be a responsive and visually appealing Full-stack application to be used by individuals via modern browsers on both desktop and mobile devices.

EventSphere enables users to plan, publish, and manage events; handle registrations with customizable forms; manage attendee participation, feedback, and certificate distribution; and track analytics. It supports role-based access for participants, organizers, and admins.

The **EventSphere** will display all upcoming and past events along with relevant details such as dates, venues, and organizing bodies. Students will have the ability to register for events online, ensuring a convenient and paperless process. Administrative users will be granted access to create, update, and delete event listings allowing for efficient content management. Additionally, the system will feature image galleries and detailed descriptions to provide a comprehensive view of each event. To keep users informed, push or email notifications will be offered for new events and important updates.

Architecture Diagram



1.5 Constraints

The **EventSphere** Web application must adhere to several constraints to ensure its successful implementation and operation. The Web application must be compatible with major Web browsers, including Chrome, Firefox, Safari, and Edge ensuring a consistent experience across different platforms. The Web application must be fully responsive, providing seamless user experience on both desktop and mobile devices. It should have fast load times and handle high traffic efficiently with real-time features such as the virtual designs performing without significant lag.

1.6 Functional Requirements

The **EventSphere** platform should be designed to support multiple user roles, each with distinct access levels and functional responsibilities. The primary types of users in the system are:

1. **Normal Student (Visitor):**
   This user is typically an unregistered or non-logged-in student who simply explores the platform. They can view public content such as upcoming events, event details, media galleries, and general pages such as About Us or Contact Us. However, they are restricted from performing actions such as event registration, feedback submission, or certificate downloads.

2. **Participant (Registered Student):**
   A participant is a student who registers on the platform to actively engage in events. This user can register for events, attend them, receive certificates (upon attendance), and provide feedback. The participant has a personalized dashboard to manage their registrations, downloads, and participation history.

3. **Organizer (College Staff):**
   Organizers are authenticated college staff members responsible for managing events within the system. They have permissions to create, edit, and schedule events. They should also manage registrations and attendance, upload media, issue certificates, and communicate with participants. Organizers help ensure that event-related processes run smoothly.

4. **Admin (System Administrator):**
   The admin user oversees the entire platform's operation. This role includes system-wide access to manage users, approve or reject events, moderate content (feedback or gallery uploads), handle role assignments, send announcements, generate detailed reports, and configure backend settings. The admin ensures overall system integrity, consistency, and data accuracy.

Functional requirements for these users are mentioned as follows:

Normal Student (Visitor)

- The user should be able to view a comprehensive list of events, including upcoming, ongoing, and past events directly from the home page or events section without the necessity to log in.
- The user should be able to browse detailed information for each event, such as the event title, description, date, time, venue, category (for example, technical or cultural), and the name(s) of organizing staff or departments.
- The user should be able to apply filters to refine the list of events based on specific categories such as department, event type (for example, seminar or competition), and date range to easily find events of interest.
- The platform should allow this user to access the media gallery, where they can view images and videos from past events, organized by event title, department, or year.
- The home page should prominently display announcements, notifications, or event banners allowing the visitor to stay informed about major or featured events.
- The user should have unrestricted access to general informational pages such as About Us, Contact Us, and FAQs, where they can learn about the platform's purpose, contact support, or find answers to common questions.
- If the user attempts to perform actions that require login—such as registering for an event, submitting feedback, or downloading a certificate—the system should prompt the user to log in or sign up, ensuring that access control is clearly communicated and enforced.

Participant (Registered Student)

- The participant should be able to register on the platform by providing necessary details such as name, email, department, and enrolment number.
- Once registered, the user should be able to log in securely using their credentials and manage their account through a personal dashboard.
- The dashboard should allow the participant to browse all available events and register for any event that is open for participation, subject to eligibility criteria and slot availability.
- The participant should receive real-time notifications and reminders related to their registered events, including schedule changes, venue

updates, and event-specific announcements.

- They should be able to cancel their registration before the cutoff date (if cancellation is permitted by event rules).
- On the day of the event, the participant should be able to check in using a system-generated QR code or other validation mechanism, which is scanned by the event organizer to record attendance.
- After successfully attending the event, the participant should be allowed to download an e-certificate from their profile or event page, upon payment of certificate fees.
- The participant can also submit feedback for events that they have attended, including star ratings and optional written comments.
- In addition, participants should be able to view their history of past events, track participation status, and bookmark events for future interest.
- They should have the option to save their favorite images or videos to their profiles for easy access later.
- Participants should have an option to pay the fees for certificate, if they require one. Payment functionality will, however, not be implemented in the project.

Organizer (College Staff)

- The organizer should be able to log in through institutional credentials or an account pre-approved by the admin with organizer-level access.
- They should be able to access an Organizer Dashboard that provides an overview of their upcoming, ongoing, and completed events as well as metrics such as registration count and feedback ratings.
- The organizer must be able to create new events by filling out comprehensive details such as event title, category, description, venue, date and time, maximum participants, and uploading media (for example, banner image or rulebook).
- Events created by organizers should enter a 'Pending Approval' state and only go live once approved by an admin.
- Organizers should be able to edit, cancel, or reschedule events prior to the event date and such changes should notify registered participants automatically.
- They should be able to monitor registrations in real time, view participant lists, and approve or reject specific registrants if required.
- On the day of the event, organizers should be able to scan QR codes or perform other authentication methods to mark attendance and also generate attendance reports post-event.
- After the event, the organizer should be able to upload certificates for

eligible participants (based on payment of certificate fees) using a pre-designed template or personalized files.

- Organizers should also be able to upload photos and videos to the gallery, moderate participant questions or feedback, and communicate directly with registered participants through announcements.

Admin (System Administrator)

- The admin should be able to log in through a secure interface with elevated credentials, possibly using Two-Factor Authentication (2FA).
- They should have access to a powerful Admin Dashboard that displays analytics such as total users by role, number of events (approved/pending), top-performing departments, and real-time system alerts.
- Admins must be able to approve, reject, or request changes to new event proposals submitted by organizers before they go live.
- They should be able to manage all users—this includes viewing user profiles, assigning roles (for example, upgrading a student to organizer), resetting passwords, suspending accounts, or deleting inactive users.
- Admins should also have the authority to moderate system content, including reviewing event descriptions, feedback entries, and media uploads to ensure quality and compliance.
- They should be able to send system-wide announcements or targeted messages to be selected roles or users, such as reminders, guidelines, or policy updates.
- Admins must be able to generate and export reports related to event participation, feedback trends, user growth, and certificate issuance in formats such as PDF or Excel.

Apart from these user specific functions, following functionalities should also be incorporated.

1. **User Registration and Login Authentication:** The registration feature will allow users to create new accounts. It will enable new users to register themselves. At the time of registration, users must provide details such as Name, Email ID, Contact Number, and Username and then, configure their password.

   Appropriate error-checking must be done on the form to ensure correct data. For example, email id can be checked to see if it is of appropriate format. (Hint: Use client-side validation).

2. **Media Gallery:** The Web application should showcase high-quality images and videos from various college events to provide a visual representation of campus life.

   The application should categorize media based on event type, date, and organizing department.

   Subcategories could include:

   - Cultural Events
   - Technical Fests
   - Sports Meets
   - Annual Day Functions
   - Workshops and Seminars
   - Intercollegiate Competitions

3. **User Dashboard:**

Under this section, the application can offer following features for registered students:

- **Event Registration Management:** Users can view, manage, and cancel their registrations for upcoming events.
- **Activity Overview:** This feature displays a summary of user's recent activity such as registered events, event participation history, and any awards or certificates received.
- **Notifications:** This feature provides alerts for upcoming events, registration deadlines, event updates, and announcements from organizers.
- **Saved Media:** Users can access and manage their collection of favorite images or videos saved from the event gallery.
- **Profile Settings:** This feature allows users to update personal details and preferences related to event participation and communication.
- **Search and Filter:** Advanced search functionality and filters to help users find events based on categories (for example, cultural, technical, or sports), date, organizing department, venue, or keyword. This allows students to quickly locate events that match their interests or schedules.

4. **User Reviews:**

Under this section, following features should be available:

- **User Type Selection**: Users should be able to select their user type from a drop-down.
- **Event Feedback Submission:** Users should be able to provide reviews and feedback for events they attended, including their overall experience, organizational quality, and content relevance.
- **Rate Event Components:** Users can rate specific aspects such as venue, coordination, technical arrangements, and hospitality.
- **Comment Section:** Users may leave comments or suggestions to help organizers understand strengths and areas for improvement.
- **View Peer Reviews:** Users can view reviews and ratings given by other participants to get insights before registering for similar future events.

5. **Dynamic Venue Capacity Management:**
   - **Event-wise configurable seating limits:** Admins can set maximum seat limits based on venue size while creating or editing an event.
   - **Automatic enforcement of capacity rules:** The system prevents further registrations once the maximum capacity is reached.
   - **Live capacity tracking and adjustments:** Admins can modify venue capacity before the event starts and availability updates accordingly.
   - **Waitlist auto-adjustment:** If a registered user cancels, the system moves waitlisted users into available slots automatically.

6. **Real-time Slot Availability:**
   - **Live slot count on event listings:** Participants can view up-to-date slot availability directly from the event list or detail pages.
   - **Auto-updates on registration/cancellation:** Slot availability adjusts immediately when users register or withdraw from the event.
   - **Visibility to promote urgency:** Real-time tracking helps encourage faster signups when slots are limited.

7. **Calendar Integration (Google, Outlook, or Apple):**
   - **Add to Calendar button on registration confirmation:** Users can save event details to their personal calendar in one click.

- **Auto-generated calendar invites in .ics format:** Includes event title, date/time, location, and reminders compatible with major calendar systems.
- **Time zone-aware syncing:** Ensures users across regions see accurate event timing in their own calendar apps.

8. **Social Media Sharing for Events:**
   - **Integrated share buttons for popular platforms:** Allows easy sharing via apps such as Facebook, WhatsApp, Twitter (X), LinkedIn, Instagram, and email.
   - **Auto-filled messages with event info:** Shared content includes event name, date, description, and registration link.
   - **Custom hashtags or promotional captions:** Admins can define reusable marketing phrases or tags for consistent branding.

9. **Certificate Fee Payment:** A feature should be added to accept fee details from participants, however actual payment processing will be beyond the scope of the project.

10. **Sitemap:** A sitemap should be created and added to the home page of the Website to help users easily understand the structure and navigation flow of the **EventSphere**.

Note: Boilerplate or readymade HTML template can be used, provided it is only for design aspect and not for implementing application functionality. Do NOT copy content or code from GPTs or other AI tools, although you are permitted to use images generated by AI tools for any visual representation purposes. It is mandatory to mention such tools used in case you add any AI generated images.

1.7 Non-Functional Requirements

There are several non-functional requirements that should be fulfilled by the application. These include:

- **Safe to use**: The application should not result in any malicious downloads or unnecessary file downloads.
- **Accessibility**: The application should have clear and legible fonts, user-interface elements, and navigation elements.
- **User-friendliness**: The application should be easy to navigate with clear menus and other elements and easy to understand.
- **Operability**: The application should be reliable and efficient.
- **Performance**: The application should demonstrate high value of performance through speed and throughput. In simple terms, the application should have minimal load time and smooth page redirection.
- **Scalability**: The application architecture and infrastructure should be designed to handle increasing user traffic, data storage, and feature expansions.
- **Security**: The application should implement adequate security measures such as authentication. For example, only registered users can access certain features.
- **Availability**: The application should be available 24/7 with minimum downtime.
- **Compatibility**: The application should be compatible with latest browsers and various devices.

These are the bare minimum expectations from the project. It is a must to implement the FUNCTIONAL and NON-FUNCTIONAL requirements given in this SRS. Once they are complete, you can use your own creativity and imagination to add more features if required.

1.8 Interface Requirements

1.8.1 Hardware

Intel Core i5/i7 Processor or higher
8 GB RAM or higher
Color SVGA monitor
500 GB Hard Disk space
Mouse
Keyboard

1.8.2 Software

Technologies to be used:

- IDE: Appropriate IDE as per the platform

- Frontend: HTML5, CSS3, Bootstrap, ReactJS/AngularJS/Angular/TypeScript, JavaScript, jQuery, and XML

- Backend: Java SDK with Apache NetBeans or Eclipse, Jakarta EE
  OR
  C# with ASP.NET MVC and ASP.NET MVC Core (optional), Visual Studio IDE
  OR
  PHP with Laravel Framework
  OR
  Python with Flask or Django
  OR
  MongoDB, Express.js, Angular, Node.js
  OR
  MongoDB, Express.js, React, Node.js
  Database: MySQL/SQL Server

Sample Database Structure

Users

Field Name	Data Type	Key	Description
user_id	INT	PK	Unique ID for the user
Email	VARCHAR(100)	UNIQUE	Email address for login
password	VARCHAR(255)		Encrypted password
Role	ENUM		User role: participant, organizer, admin
created_at	DATETIME		Account creation timestamp

UserDetails

Field Name	Data Type	Key	Description
detail_id	INT	PK	Unique detail record ID
user_id	INT	FK	References Users(user_id)
full_name	VARCHAR(100)		Full name of the user
mobile	VARCHAR(15)		Mobile contact number
department	VARCHAR(100)		User's academic department
enrollment_no	VARCHAR(50)		Student enrollment number

Events

Field Name	Data Type	Key	Description
event_id	INT	PK	Unique ID for the event
Title	VARCHAR(150)		Title of the event
description	TEXT		Detailed event description
category	VARCHAR(50)		Event type: technical, cultural, and so on.
Date	DATE		Event date
Time	TIME		Event time
Venue	VARCHAR(100)		Event location
organizer_id	INT	FK	References Users(user_id)

Registrations

Field Name	Data Type	Key	Description
registration_id	INT	PK	Unique registration record
event_id	INT	FK	References Events(event_id)
student_id	INT	FK	References Users(user_id)
registered_on	DATETIME		Timestamp of registration
Status	ENUM		confirmed, cancelled, waitlist

Attendance

Field Name	Data Type	Key	Description
attendance_id	INT	PK	Unique attendance record
event_id	INT	FK	References Events(event_id)
student_id	INT	FK	References Users(user_id)
attended	BOOLEAN		TRUE if attended, FALSE if absent
marked_on	DATETIME		Timestamp of attendance marking

Feedback

Field Name	Data Type	Key	Description
feedback_id	INT	PK	Unique feedback record
event_id	INT	FK	References Events(event_id)
student_id	INT	FK	References Users(user_id)
Rating	INT		Rating from 1 to 5
comments	TEXT		Optional written feedback
submitted_on	DATETIME		Timestamp of feedback submission

Certificates

Field Name	Data Type	Key	Description
certificate_id	INT	PK	Unique certificate record
event_id	INT	FK	References Events(event_id)
student_id	INT	FK	References Users(user_id)
certificate_url	VARCHAR(255)		File path or URL of the certificate
issued_on	DATETIME		Date certificate was issued

MediaGallery

Field Name	Data Type	Key	Description
media_id	INT	PK	Unique media file ID
event_id	INT	FK	References Events(event_id)
file_type	ENUM		Type of file: image, video
file_url	VARCHAR(255)		File path or URL
uploaded_by	INT	FK	References Users(user_id)
caption	VARCHAR(150)		Optional media caption
uploaded_on	DATETIME		Timestamp of upload

Event Seating

Field Name	Data Type	Description
event_id	INT (PK, FK)	References the event
venue_id	INT (FK)	References the venue where the event is held
total_seats	INT	Total seats assigned for the event
seats_booked	INT	Number of seats already booked
seats_available	INT (derived)	Total - booked; auto-calculated
waitlist_enabled	BOOLEAN	If true, system adds users to waitlist when event is full

Event Waitlist

Field Name	Data Type	Description
waitlist_id	INT (PK)	Unique ID for waitlist entry
user_id	INT (FK)	ID of the waitlisted user
event_id	INT (FK)	Event they're waitlisted for
waitlist_time	DATETIME	Timestamp when the user was added to waitlist
status	ENUM	Values: waiting, confirmed, cancelled

Calendar Sync

Field Name	Data Type	Description
sync_id	INT (PK)	Unique calendar sync ID
user_id	INT (FK)	User who synced the event
event_id	INT (FK)	Event synced to calendar
calendar_type	VARCHAR	Google, Outlook, Apple, and so on.
sync_timestamp	DATETIME	When sync was triggered
calendar_url	VARCHAR	URL to the .ics file or calendar entry

Event Share Log

Field Name	Data Type	Description
share_id	INT (PK)	Unique ID of the share action
user_id	INT (FK)	ID of the user who shared the event
event_id	INT (FK)	Event being shared
platform	VARCHAR	Platform used: Facebook, WhatsApp, and so on.
share_timestamp	DATETIME	Date and time of sharing
share_message	TEXT	Auto-generated message content

Note: These are just examples; you do not have to adhere to these structures and can design your own table structure with more or less columns.

1.9 Project Deliverables

You will design and build the project and submit it along with a complete documentation that includes:

- Problem Definition
- Design Specifications
- Diagrams such as Flowcharts for various Activities, Data Flow Diagrams, and so on
- Database Design
- Test Data Used in the Project
- Project Installation Instructions
- **User Credentials for all Types of Users with Passwords**

Documentation is considered as a very important part of the project. Ensure that documentation is complete and comprehensive. However, do not include full source code inside the documentation.

The consolidated project must be submitted as a zip file with a ReadMe.doc file listing assumptions (if any) made at your end and SQL scripts files (.sql) containing database and table definitions.

Note: Preferably, host the working Web application on a Website and share the URL for evaluation.

Submit a video (.mp4 file) demonstrating the working of the full-stack Web application, including all the functionalities of the project. This is MANDATORY.

Sitemap: To help users understand the flow of the Web application, you will create a Sitemap and add it to the home page of your Web application.

Over and above the given specifications, you can apply your creativity and logic to improve the application.

~~~ End of Document ~~~