TECHWIZ 6

GLOBAL AI-BASED TECH COMPETITION

Aptech

Unleash your potential

Software Requirements Specification

Version 1.0

Project Name: CampusConnect

Theme: College Event Information System
Category: Responsive NextGen Website Development

Table of Contents

1.1 Background and Necessity for a Website

College campuses are hubs of student activity and creativity. Events such as technical fests, cultural festivals, workshops, and competitions are a regular part of student life.

Students and staff often seek information on the following:

- Upcoming events and their schedules
- Information about past events
- Types of different events
- Images/media galleries of past events
- Department-wise listing of events

However, as of today, there is often a lack of centralized communication and visibility of these events.

There is a necessity for a Website to bridge this gap by providing a centralized portal where students and faculty can view information about upcoming, ongoing, and past events organized within the college and gain other related data about them.

1.2 Proposed Solution

The solution is to develop a responsive Website titled **CampusConnect**, which displays categorized information about events. The Website will read event data from JSON files and display it through a clean, well-structured UI. It will support viewing department-wise events, category-wise filtering (technical, cultural, or sports), event descriptions, schedules, organizers, and media gallery.

It will serve as a centralized hub for students and faculty to explore upcoming events and access related details.

The target audience for this Website includes:

- Students: Looking for information on upcoming events, their interests, and registration details.
- Faculty: Wishing to promote events organized by their departments or clubs.
- Visitors: Interested in college events open to the public.

1.3 Purpose of the Document

The purpose of this document is to explain the features of the Website, interfaces of the Website, what the Website will do, and the constraints under which it must operate. This document is intended for both stakeholders and developers of the Website.

1.4 Scope of the Project

The **CampusConnect** portal will feature responsive user interface designed for clarity and ease of use. It will dynamically read event details—including title, description, date, department, type, image path, and organizer information — from a JSON data source. Users will be able to filter and search for better accessibility. While the feedback form included will be non-functional, it will serve as a UI demonstration element. The portal will consist of multiple pages such as **Home**, **About**, **Gallery**, **Events**, **Feedback**, and **Contact** all designed with a focus on visual consistency and user-friendly navigation.

1.5 Constraints

The **CampusConnect** portal is designed as a purely client-side application with no backend or server-side functionality. It will rely on **read-only JSON data** for displaying event details, ensuring simplicity and ease of deployment. The feedback form included will be static and non-functional, serving only as a visual element. Additionally, the project will not incorporate any third-party design templates, maintaining a fully original and custom-built interface.

1.6 Functional Requirements

The portal will consist of following core functional components:

Home Page:

The **Home Page** acts as the primary landing page, hence it should be offering a clear, visually engaging introduction to ***CampusConnect***. It should inform users about current happenings, direct them to relevant sections, and establish the platform's purpose and usability from the very first interaction.

- The home page displays welcome message and banner:

  - A prominent **welcome message** should be shown at the top of the page, welcoming users (students, staff, and guests) to the ***CampusConnect***.
  - The welcome message can be like *"Welcome to [College Name] Event Hub – Stay Updated, Stay Involved!"* to give users an immediate sense of what the site offers.
  - A **banner image or slideshow** should be placed just below or alongside the welcome message, featuring college branding, event snapshots, or seasonal celebrations.
  - The banner may be implemented using HTML/CSS or simple JavaScript sliders to add visual engagement without relying on a database.
  - Banner images and messages can be read from a static JSON file for dynamic loading without server interaction.
- The home page displays navigation menu with links to different sections.

- Also, a clean and responsive **navigation bar** should appear at the top of the page (or as a side menu for smaller screens), allowing users to quickly explore the Website.
- Typical menu links include:
  - Home
  - About Us
  - Event Calendar
  - Event Details
  - Registration (Static form/link)
  - Gallery
  - Contact Us

- The home page displays highlights of upcoming events.

- This section should **brief the viewers on highlights of important or upcoming events**, such as:
  - 'TechFest 2025 – July 20'
  - 'Cultural Week Begins – August 5'
- The site should show data for upcoming events (title, date, and short description) is fetched from a **local JSON file** and displayed dynamically using JavaScript.
- The section may include **event thumbnails, countdown timers, or 'Learn More' buttons** that navigate to the event's dedicated detail page.
- For visual appeal, **cards or carousels** can be used to display event summaries in a grid or slider format.

• Responsive layout should be used with attractive animations.

About Page

The **About Page** provides viewers with essential background information about the college, its traditions, the types of events it hosts, and the teams or bodies that organize these events. This page builds context and trust, especially for new students, guests, or potential sponsors.

• Displays information about the college, key annual events, and organizing bodies

- This section offers a brief, but informative **overview of the**

college, such as:

- Name, location, and affiliations (For example, 'XYZ College of Engineering, affiliated with ABC University')
- Campus highlights or recognitions

➢ A subsection outlines the **key annual events** hosted by the college:

- **Technical events** such as coding marathons, project exhibitions (for example, *TechFest, Hackathon, or Robotics Championship*)
- **Cultural events** such as *Annual Day, Music Nights, or Dance Competitions*
- **Sports and other activities** such as *Inter-college Sports Meet, Blood Donation Drives, or Alumni Meet*

➢ Brief descriptions and month-wise timelines may be shown to help users understand the scale and frequency of events.

Event Listing Section/Event Catalog

- The system should **display a structured list of upcoming and past events**, where each event entry includes key details such as the **event name, date, time, venue, and a short description**. This information will be loaded dynamically from a local JSON file and rendered on the Web page using JavaScript.
- Allow filtering and sorting events by category (for example, academic, cultural, or sports).
- Each event item in the list should appear in a **uniform card or tabular layout**, allowing users to quickly view essential event information at a glance. For example, an event card should display:

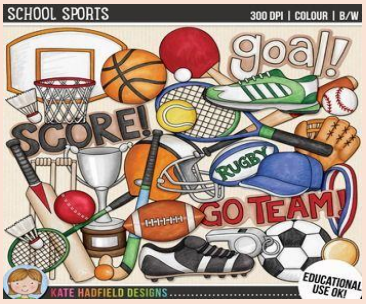
➢ **Event Name**: Prominently as the title

➢ **Date and Time**: In a clear, readable format (for example, "20 Sep 2025, 10:00 AM – 4:00 PM")

➢ **Venue**: Location of the event (for example, "Main Auditorium", "Lab 302")

➢ **Brief Description**: A short summary (2–3 lines) explaining the purpose or theme of the event.

- The system should provide **filtering options** that allow users to view events by specific **categories** such as:



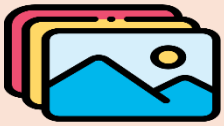
  - Academic Events
  - Cultural Events
  - Sports Events
  - Departmental Events

  These categories should be defined in the JSON data and selectable via a drop-down or tab-based menu on the Web page.

- Additionally, the system should offer **sorting functionality**, enabling users to organize the event list by:
  - Date (Upcoming or Most Recent First)
  - Event Name (Alphabetical Order)
  - Category

Gallery Page

- The system shall display event-related images through a JSON data file. Ensure that no server-side image hosting or processing is required, keeping the site lightweight and easy to maintain.
- Display images organized by year or category.



The Gallery page shall allow users to view event images grouped by academic year (for example, 2022–23, 2023–24, and so on) or by category (such as Technical, Cultural, Sports, and so on.). A filter mechanism (drop-down or tabs) shall be implemented to rearrange the displayed content based on the selected grouping. This ensures intuitive navigation and allows users to explore past events efficiently.

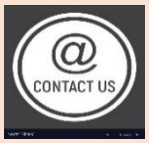
Feedback Page

• Feedback form for UI purpose

The **Feedback page** will include a **feedback form** that is designed purely for user interface demonstration purposes and will not perform any form submission or data processing. This form will serve as a visual representation of a typical feedback mechanism to showcase design and layout skills.

The form will include the following input fields:

- **Name:** A text input field for the user to enter their full name.
- **Email:** A field to input a valid email address, typically used for user identification or response (though not validated or stored).
- **User Type:** A drop-down or text field specifying user type or role, for example, 'Student', 'Faculty', and so on.
- **Event Attended:** A drop-down or text field allowing the user to specify the event they wish to give feedback on (limited to past one month only).
- **Rating:** A 1–5 scale rating system (could be a drop-down, radio buttons, or star icons) to reflect their experience with the event.
- **Comments:** A multiline text area where users can provide additional remarks, suggestions, or opinions.



Contact Page

• Displays contact details of faculty and student coordinators:
The Contact page shall present a list or card-style layout showing the names, designations, departments, phone numbers, and email addresses of faculty members and student coordinators involved in organizing campus events. This data may be fetched from a JSON file and displayed using Angular or ReactJS components. The layout will be responsive and structured for easy readability across devices.

• Google Map embed:
The Contact page shall include an embedded Google Map element showing the location of the college campus. The embed will use a location pin via Google Maps' sharing feature, helping users visually locate the venue for offline or physical events. The map will be integrated as a fixed-width responsive component, ensuring consistent layout across

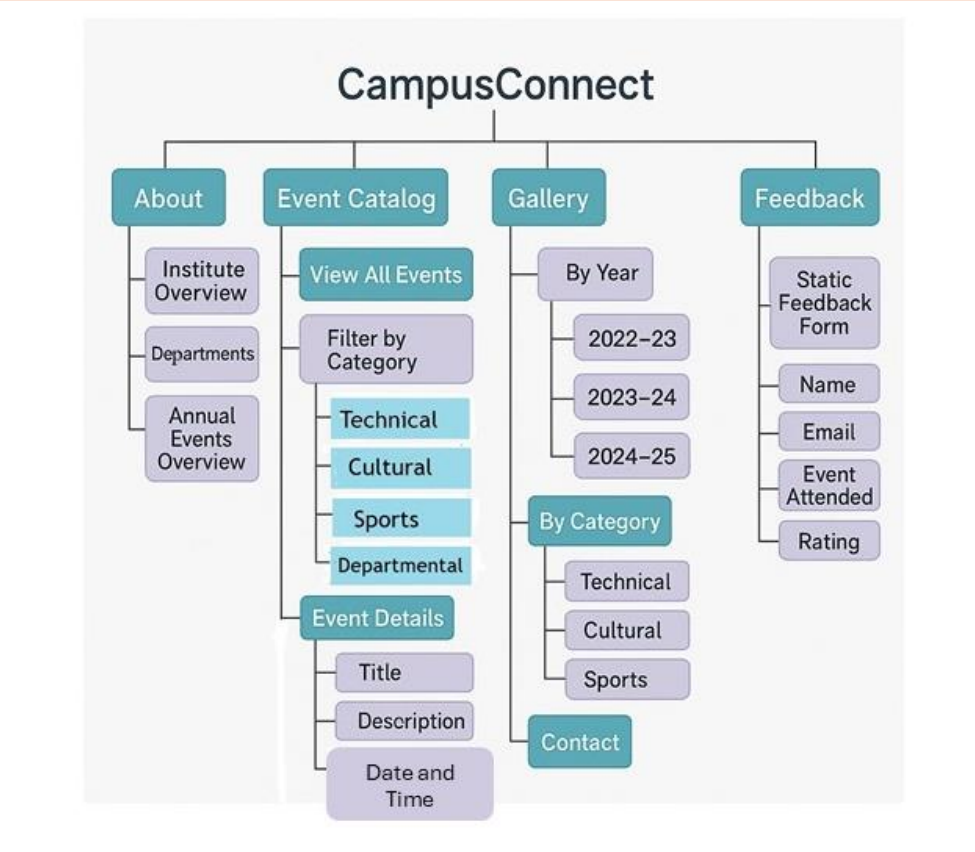
screen sizes.

Bookmarking

This may not be a menu option, but should be implemented as a feature.

Images/text content can be favorited or bookmarked (temporarily for the session duration only). You may make use of local storage or IndexedDB to implement this.

Sample Sitemap



Technical Considerations:

Use HTML5, CSS3, Figma UI Toolkit, Bootstrap, jQuery, and JavaScript to build the User Interface (UI).

Use either Angular or ReactJS to develop the application's frontend and add dynamic and responsive features along with SPA functionality.

Use JSON/TXT files to handle data retrieval.

Ensure that the application is responsive and compatible with different screen sizes and browsers.

Tasks:

Design **CampusConnect** system to present a user-friendly interface that allows users to browse and explore campus events efficiently. Key tasks include displaying an event catalog, providing a search bar to locate events by name or category, and offering sorting options based on date, department, or popularity. The system will also feature an event information section that presents detailed descriptions. Supporting services will handle search, sorting, and filtering operations to ensure smooth interaction and relevant results for users.

Implement GPS functionality as applicable.

Optionally, use REST APIs if you are well-versed with them.

Test the Website's functionality including browsing and searching. Deploy the Website to a local Web server such as XAMPP for testing purposes.

Important: Do NOT use boilerplate templates or readymade templates for development as it will adversely affect your evaluation. Your own Website design and development skills will be tested; hence no third-party templates should be used here. 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 Website.

Some of these include:

Safe to use: The Website should not result in any malicious downloads or unnecessary file downloads.
Accessibility: The Website should have clear and legible fonts, user-interface elements, and navigation elements.
User-friendliness: The Website should be easy to navigate with clear and easy to understand elements.
Operability: The Website should be reliable and efficient.
Performance: The Website should demonstrate high value of performance through speed and throughput. In simple terms, the Website should have minimal load time and smooth page redirection.
Scalability: The Website should support as large number of concurrent users.
Availability: The Website should be available 24/7 with minimum downtime.
Compatibility: The Website should be compatible with the 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:

Frontend: HTML5, CSS3, Bootstrap (optional), JavaScript, Figma Toolkit, jQuery, AngularJS/Angular 9 or higher/ReactJS 18 or higher, and XML

IDE: Visual Studio Code/Notepad++/CoffeeCup or any other HTML Editor

Data Store: JSON/TXT

Local Testing Server: XAMPP

1.9 Project Deliverables

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

- Problem Definition
- Design Specifications
- Diagrams such as Flowcharts for various Activities, Data Flow Diagrams, and so on
- Test Data Used (if any)
- Project Installation Instructions

The consolidated project must be submitted as a zip file containing the source code used for the complete Website. Include a ReadMe file mentioning software used, assumptions made (if any), and test data used.

Ensure that documentation is complete and comprehensive.

Documentation should not contain any source code.

Submit a video (.mp4 file) demonstrating the working of the Website, including all the functionalities of the project. This is MANDATORY.

Optionally, a live hosted URL can be supplied for the Website and mention the URL in the documentation or ReadMe file. Over and above the specifications given, you can apply your creativity and logic to improve the portal.