



NEW HORIZON COLLEGE OF ENGINEERING

Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

A MINI PROJECT REPORT

on

WEB DESIGN TECHNOLOGIES (24CSE361)

COLLEGE FEST WEBSITE

Submitted by

**NAME:M Prajval
USN:1NH24CS118**

Under the guidance of
Ms. Divyanshi Chhabra
Assistant Professor

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

Academic Year: 2025-26 (ODD SEM)



NEW HORIZON

COLLEGE OF ENGINEERING

Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA

CERTIFICATE

This is to certify that the mini project work titled "**College Fest Website**" is a bonafide work carried out by M Prajval (**1NH24CS118**) in partial fulfillment of the degree of **Bachelor of Engineering in Computer Science and Engineering** of the New Horizon College of Engineering during the year **2025-2026**.

Signature of Guide

Signature of HOD

SEMESTER END EXAMINATION

Name of the Examiner

Signature with date

1. _____

2. _____

ABSTRACT

The College Fest Website is a dynamic and user-friendly platform developed to provide students, faculty, and visitors with complete details of the annual college fest in an organized and interactive manner. Built using HTML, CSS, and JavaScript, the system focuses on delivering a visually appealing layout along with smooth navigation and responsive behavior across all devices. The website includes key modules such as event listings, schedules, online registration, contact details, and a multimedia gallery, ensuring that users can easily explore fest activities and participate in various competitions.

The design integrates modern web-development practices, including mobile responsiveness, optimized images, and interactive components like countdown timers, event filters, and lightbox galleries to enhance user engagement. The system allows fest organizers to update event details quickly, making it adaptable for annual use. With its clean interface, fast performance, and accessibility-driven structure, the website provides a complete digital solution for showcasing the fest's highlights and improving communication between organizers and participants.

The project demonstrates the practical application of web technologies in solving real-world communication challenges within academic events. By combining aesthetics, usability, and interactivity, the College Fest Website serves as an efficient, modern, and scalable platform for managing and promoting college-level cultural and technical festivals.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, who's constant guidance and encouragement crowned our efforts with success.

I have great pleasure in expressing gratitude to **Dr. Mohan Manghnani**, Chairman, New Horizon Educational Institutions, for providing necessary infrastructure and creating good environment.

I take this opportunity to express my profound gratitude to **Dr. Manjunatha**, Principal, New Horizon College of Engineering, for the constant support and encouragement.

I would like to thank **Dr. Anandhi R J**, Professor and Dean-Academics, NHCE, for her valuable guidance.

I would also like to thank **Dr. B. Rajalakshmi**, Professor and HOD, Department of Computer Science and Engineering, for the constant support.

I also express my gratitude to **Ms.Divyanshi Chhabra**, Assistant Professor , Department of Computer Science and Engineering, my mini project reviewer, for constantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing the work.

M Prajval

1NH24CS118

CONTENTS

ABSTRACT	I
ACKNOWLEDGEMENT	II
CONTENTS	III
LIST OF FIGURES	V
1. INTRODUCTION	1
1.1 PROBLEM DEFINITION	1
1.2 OBJECTIVES	2
1.3 METHODOLOGIES TO BE FOLLOWED	3
2. FUNDAMENTALS OF THE LANGUAGES USED	4
2.1 HTML	4
2.2 HTML TAGS	6
2.3 CSS	6
2.4 JAVASCRIPTS	7
3. REQUIREMENT SPECIFICATION	11
3.2 SOFTWARE REQUIREMENTS	11
4. DESIGN	12
4.1 DESIGN GOALS	12
5. IMPLEMENTATION	13
5.1 HOME PAGE	13
5.2 PROFILE PAGE	15

5.3	SIGNUP AND LOGIN PAGE	16
6. RESULTS		17
6.1	MAIN PAGE	17
6.2	PROFILE PAGE	19
6.3	SIGNUP AND LOGIN PAGE	20
7. CONCLUSION		20
REFERENCES		21

LIST OF FIGURES

Figure No.	Description	Page No.
6.1	Home page	25
6.2	Event line-up	25
6.3	Past event highlights	26
6.4	Past event highlights slide-2	26
6.5	Mystery guest Reveal Page	27
6.6	Campus Snapshots	27

LIST OF TABLES

Table No.	Description	Page No.
2.2	HTML TAGS	14
2.3	CSS TAGS	15,16

CHAPTER 1

INTRODUCTION

1.1 PROBLEM DEFINITION

College festivals play a vital role in showcasing students' talents, creativity, and technical skills while bringing the entire campus together. As these events continue to grow in scale and participation, the need for an organized, informative, and easily accessible communication platform becomes essential. Traditional methods of sharing fest details—such as posters, announcements, and manual registrations—are often inefficient and fail to reach the entire student community on time. To address this gap, a dedicated College Fest Website provides a modern digital solution.

The College Fest Website is designed to offer a centralized platform where users can explore event categories, view schedules, register for competitions, and access important updates in a seamless manner. By using standard web technologies like HTML, CSS, and JavaScript, the website ensures fast loading, responsive behavior, and an engaging user experience. Features such as an interactive gallery, countdown timer, lightbox image view, and filterable event lists enhance both usability and visual appeal.

This project aims not only to digitize the event information but also to streamline the overall fest management process. Participants can register online, organizers can update information efficiently, and visitors can explore the fest without confusion. The website thereby solves communication challenges, supports event coordination, and promotes the fest in a more professional and widespread manner. Overall, the system reflects the practical implementation of modern web development concepts in an academic and real-world context.

1.2.OBJECTIVE

The main objectives of developing the College Fest Website are as follows:

1. To provide a centralized digital platform

Create a single online space where students, faculty, and visitors can easily access all fest-related information such as events, schedules, rules, and updates.

2. To simplify event registration

Enable users to register for multiple events online without manual paperwork, thereby reducing errors and improving participation management.

3. To enhance communication and visibility

Offer real-time announcements, highlights, and updates so that participants stay informed about fest activities at all times.

4. To ensure smooth user navigation

Design an interface that is easy to use, intuitive, and well-structured, allowing users to explore the website quickly without confusion.

5. To provide an engaging experience

Include interactive features such as image galleries, countdown timers, event filters, and pop-up previews to make the website more dynamic and attractive.

6. To support responsive access

Ensure that the website works effectively on mobile phones, tablets, and laptops, providing consistent performance across all devices.

7. To allow easy maintenance

Create a backend-friendly structure (even if manually updated) so organizers can quickly change event details, images, and schedules each year.

8. To promote the college fest professionally

Present the fest in a modern, visually appealing format that reflects the college's identity and attracts a wider audience.

9. To improve communication between organizers and participants

Provide clear instructions, contact details, and event descriptions that reduce confusion and miscommunication.

10. To enhance user engagement through interactive features

Integrate countdown timers, gallery pop-ups, and filter systems to make the interface lively and engaging.

11. To ensure platform independence

Design the website so it works smoothly on all operating systems (Windows, macOS, Android, iOS) without special dependencies.

12. To enable easy navigation through a clean UI

Implement intuitive menus, smooth scrolling, and clearly labeled sections for seamless user navigation.

13. To allow quick updates to events and schedules

Use dynamic JavaScript functions so organizers can modify event details without editing entire pages.

14. To promote responsive web design principles

Ensure the website looks and functions properly on devices of all screen sizes.

15. To maintain consistency across all pages

Follow uniform formatting, colors, spacing, and typography to provide a polished and professional appearance.

16. To create a secure registration interface

Implement basic validation to ensure users enter correct information and avoid data-entry mistakes.

17. To demonstrate practical application of web technologies

Show how HTML, CSS, and JavaScript can be integrated to build a real-world, functional website.

18. To showcase the college fest in an attractive digital format

Present events, images, and key highlights in a visually engaging way to increase student participation.

19. To simplify the registration process for users

Provide a fast, user-friendly form that allows participants to register for multiple events without confusion.

20. To ensure smooth interaction using lightweight scripts

Use optimized JavaScript so that animations, filters, and navigation run efficiently even on low-end devices.

1.3.METHODOLOGIES TO BE FOLLOWED

The development of the College Fest Website followed a structured methodology to ensure that the final system is user-friendly, efficient, and visually appealing. The following steps outline the methodological approach taken:

1. Requirement Analysis

- Collected requirements such as event listing, schedule display, gallery, registration form, and contact details.
- Understood the target users—students, organizers, and visitors.
- Identified functional and non-functional requirements.

2. Planning and Design

- Created a layout plan for all sections including Home, Events, Gallery, Schedule, Registration, and Contact.
- Designed the wireframe for an intuitive flow.
- Selected a color theme, fonts, and overall design approach.
- Planned a responsive layout that adjusts to all screen sizes.

3. Front-End Development

- Implemented the structure using HTML.
- Designed the visual appearance with CSS, including grids, cards, colors, spacing, and formatting.
- Added interactive features using JavaScript such as:
 - Countdown timer
 - Event filtering
 - Image pop-up modal
 - Form validation
 - Smooth scrolling
 - Local storage for demo registration
- Ensured responsiveness using flexible layouts.

4. Content Integration

- Inserted event details, schedule data, images, gallery items, and registration options.
- Replaced placeholder images with actual images according to fest requirements.
- Ensured content is clean, readable, and well-structured.

5. Testing

- Tested the website for:
 - Functionality (forms, filters, countdown, buttons)
 - Responsiveness on mobile, tablet, and desktop
 - Performance and loading time
 - Browser compatibility
- Checked for errors, dead links, and alignment issues.

6. Optimization

- Optimized code by separating HTML, CSS, and JavaScript.
- Reduced image sizes for faster loading.
- Ensured proper accessibility (alt tags, readable fonts, good contrast).

7. Deployment and Maintenance

- Prepared the website files for hosting.
- Tested the site in a live environment.
- Ensured the structure allows organizers to update event details every year with minimal changes.

CHAPTER 2

FUNDAMENTALS OF THE LANGUAGES USED

2.1 HTML

HTML is the foundational markup language used to create and structure web pages on the World Wide Web. It defines the layout of a webpage by using elements enclosed within tags such as `<h1>`, `<p>`, and `<div>`. HTML was originally developed by Tim Berners-Lee in 1989 as part of the World Wide Web project at CERN, helping researchers share documents through hyperlinked text. The earliest version of HTML, introduced in 1991, contained just 18 basic tags.

HTML has evolved significantly over time. In 1995, HTML 2.0 became the first standardized version and introduced essential features such as forms and tables. HTML 3.2 (1997) expanded support for styling through CSS, while HTML 4.01 (1999) added scripting support and improved structural markup. In the early 2000s, XHTML attempted to bring stricter XML rules to web development but faced adoption challenges. The release of HTML5 in 2014 transformed modern web development by introducing semantic tags, built-in multimedia support, canvas elements, and improved cross-platform compatibility. Today, HTML remains the backbone of every website on the internet.

Key Features:

- Structure: Organizes content using headings, paragraphs, lists, images, and sections.
- Hyperlinking: Allows navigation across documents using `<a>` tags.
- Forms: Collects user input through text fields, buttons, checkboxes, and other controls.
- Semantic Elements: Tags like `<header>`, `<article>`, `<section>`, and `<footer>` improve clarity and accessibility.
- Media Support: Embeds audio, video, and graphics without external plugins.

2.2 HTML TAGS

TAG	DESCRIPTION
<!DOCTYPE HTML>	Declares that the document is an HTML5 file.
<HTML>	The root element of the webpage; all content is written inside it.
<HEAD>	Contains metadata, page title, and links to CSS and JS files.
<META>	Defines metadata such as character set and viewport settings.
<TITLE>	Sets the title of the webpage (shown on browser tab).
<LINK>	Links external CSS file (style.css) to the HTML document.
<BODY>	Contains the visible content of the webpage.
<HEADER>	Defines the top section of the page, usually for titles and introductions.
<H1>	Defines the main heading (largest heading).
<H2>, <H3>	Define subheadings and smaller titles for sections or cards.
<P>	Defines a paragraph of text.
<NAV>	Represents the navigation section (menu bar).
	Defines an unordered list (used for navigation menu).
	Defines list items inside the unordered list.
<A>	Defines a hyperlink (used for navigation links).
<SECTION>	Groups related content together (like Home, Events, Gallery, Contact).
<DIV>	Generic container used for grouping elements and styling with CSS.
	Embeds an image in the webpage.
<FORM>	Defines an input form for user data (like contact form).
<LABEL>	Describes an input field.
<INPUT>	Creates a textbox or email box for user input.
<TEXTAREA>	Creates a larger text box for typing messages.
<BUTTON>	Creates a clickable button (for submitting forms).
<FOOTER>	Defines the footer section at the bottom of the page.
<SCRIPT>	Links or embeds JavaScript code (here it connects to script.js).

2.3.CSS

CSS is the stylesheet language used to control the presentation, layout, and visual appearance of HTML documents. While HTML defines the structure of a webpage, CSS determines how that structure looks—its colors, fonts, alignment, spacing, borders, animations, and overall design. CSS was first introduced by Håkon Wium Lie in 1994 while working with Tim Berners-Lee at CERN. The goal was to separate content from design and allow designers more control over webpage styling.

The first official specification, CSS1, was released in 1996, providing basic features such as fonts, colors, and text formatting. CSS2, introduced in 1998, expanded capabilities by adding positioning, z-index, media types, and improved layout control. With the growth of modern, dynamic websites, CSS evolved into CSS3, which was released in modules starting from 2011. CSS3 introduced major advancements such as transitions, animations, shadows, flexbox, grid layout, and responsive design features—greatly improving how modern websites look and behave. Today, CSS is essential for creating responsive, interactive, and visually rich web experiences.

Key Features:

- Styling and Formatting: Controls colors, backgrounds, borders, fonts, and spacing.
- Layout Management: Uses flexbox, grid, floats, and positioning to arrange page elements.
- Responsive Design: Adapts webpages to different screen sizes using media queries.
- Reusability: A single stylesheet can style multiple webpages, reducing duplication.
- Animations and Effects: Provides transitions, keyframe animations, shadows, and transformations.

• CSS PROPERTY	DESCRIPTION
BACKGROUND	Sets background color, gradient, or image.
BACKGROUND-COLOR	Applies solid background color.
BACKGROUND-IMAGE	Adds image or gradient backgrounds.
COLOR	Sets text color.
FONT-FAMILY	Sets the font style for text.
FONT-SIZE	Sets the size of text.
FONT-WEIGHT	Defines text thickness (bold, normal).
MARGIN	Creates space outside an element.
PADDING	Creates space inside an element.

BORDER	Adds border around elements.
BORDER-RADIUS	Rounds the corners of elements.
BOX-SHADOW	Adds shadow effect around cards and buttons.
WIDTH	Sets the width of an element.
HEIGHT	Sets the height of an element.
DISPLAY	Controls how an element behaves (block, flex, grid, inline).
FLEX	Allows flexible layout inside flex containers.
GAP	Controls spacing between flex or grid items.
JUSTIFY-CONTENT	Aligns items horizontally inside flex containers.
ALIGN-ITEMS	Aligns items vertically in flex containers.
GRID-TEMPLATE-COLUMNS	Defines column structure in grid layouts.
OBJECT-FIT	Controls how images adjust within containers.
CURSOR	Changes the cursor icon (e.g., pointer).
POSITION	Defines element positioning (relative, absolute, fixed).
TOP, LEFT, RIGHT, BOTTOM	Positions elements with position property.
Z-INDEX	Controls stacking order of elements.
OVERFLOW	Handles content overflow (hidden, scroll).
TEXT-ALIGN	Aligns text (left, center, right).
LIST-STYLE	Removes or modifies list bullets.
TRANSITION	Adds smooth animation effects.
OPACITY	Controls transparency of elements.
FILTER	Applies visual effects (blur, brightness).
LINE-HEIGHT	Sets spacing between lines of text.
MAX-WIDTH	Sets maximum width of elements.
MIN-WIDTH	Sets minimum width.
BOX-SIZING	Defines how width and height are calculated.

2.3 JAVASCRIPT

JavaScript is a high-level, lightweight, and versatile programming language primarily used to create interactive and dynamic web applications. It enables developers to add features such as animations, form validations, and real-time content updates. Initially developed by Brendan Eich at Netscape in 1995, it was originally called Mocha, later renamed LiveScript, and eventually JavaScript to align with the popularity of Java at the time. JavaScript gained widespread adoption with the introduction of ECMAScript standards in 1997, which ensured consistency across implementations. Over the years, JavaScript has evolved significantly, adding modern features like asynchronous programming, modularization, and extensive libraries, making it a cornerstone of modern web development alongside HTML and CSS.

Key Features and examples:

- **Basics:** Understanding variables (var, let, const), data types, operators, and expressions.
- **Functions:** Writing reusable blocks of code to perform specific tasks.
- **DOM Manipulation:** Selecting and modifying elements dynamically.
- **Event Handling:** Responding to user actions like clicks and keypresses.
- **Form Validation:** Ensuring proper user input.

CHAPTER 3

REQUIREMENT SPECIFICATION

SOFTWARE REQUIREMENTS:

1. Operating System

- Windows 10 / Windows 11
- macOS
- Linux (Ubuntu recommended)

Any OS that supports a web browser and code editor can run the project.

2. Web Browser

A modern browser is required to view and test the website:

- Google Chrome (recommended)
- Mozilla Firefox
- Microsoft Edge
- Safari

These browsers support HTML5, CSS3, and JavaScript required for the website.

3. Code Editor / IDE

To create and edit the website files:

- Visual Studio Code (recommended)
- Sublime Text
- Notepad++
- Atom
- Brackets
- Basic Notepad (for simple editing)

4. Languages/Techologies Used

- HTML5 – for creating the structure and layout of the website
- CSS3 – for styling and design
- JavaScript (ES6) – for interactivity and dynamic features

5. Optional Tools (for better workflow)

- Live Server Extension (VS Code) for real-time preview
- Git & GitHub for version control
- Browser DevTools (Inspect Element) for debugging CSS/JS
- Image optimization tools (e.g., TinyPNG)

6. Runtime Environment

No special environment required—

The project runs directly in the browser without installation.

7. Additional Libraries (If needed)

- Google Fonts (for typography)
- Icons or graphic resources (Freepik/Icons8)

CHAPTER 4

DESIGN

4.1 DESIGN GOALS

The main design goals of the College Fest Website focus on creating a platform that is visually appealing, user-friendly, and functionally efficient. The intention is to provide students, organizers, and visitors with a seamless browsing experience while showcasing the events, schedule, and registration features clearly.

1. User-Friendly Interface

The website should be easy to navigate, with a clean layout and clear content organization. Users must be able to find information such as events, schedules, gallery, and registration without confusion.

2. Responsive Design

The site must work efficiently on all devices—mobile phones, tablets, and desktops. Responsive layouts ensure accessibility for all users.

3. Attractive Visual Presentation

Through the use of colors, images, animations, and layout techniques, the website should create excitement and capture the essence of a vibrant college fest.

4. Fast Performance

The pages should load quickly with optimized images and efficient code so users do not face delays.

5. Easy Event Registration

A simple and intuitive registration form should allow users to sign up for events effortlessly. Validation should ensure accurate data entry.

6. Dynamic Content

The website should dynamically update elements such as event cards, countdown timer, and gallery images using JavaScript to create an interactive user experience.

7. Consistency Across Pages

Fonts, colors, spacing, and layouts must remain consistent across the entire website to maintain a professional look.

8. Accessibility

Content should be readable, buttons should be clearly labeled, and navigation should be smooth so the website is usable for all types of users.

9. Easy Maintenance and Scalability

The website should be coded in a modular way so events, images, and schedules can be easily updated without rewriting the whole website.

10. Security and Privacy

The website should ensure that user information submitted through forms (name, email, year, message) is handled safely. Even though this is a demo, the structure should follow good practices to protect user data.

11. Modular Code Structure

HTML, CSS, and JavaScript should be separated into different files so the codebase remains clean, readable, and easy for future developers to enhance.

12. Search and Filter Efficiency

The events filtering system should work smoothly without page reloads, allowing users to quickly sort events by category.

13. Cross-Browser Compatibility

The website must function consistently across major web browsers such as Chrome, Firefox, Safari, and Edge. No design or functional issues should break the experience.

14. Engaging Interactivity

Animations, hover effects, modal image viewer, and countdown timer should all contribute to a more engaging user experience, encouraging visitors to explore more sections.

15. Maintainability

The website should be easy to update. Admins or developers must be able to change events, schedules, and gallery images without modifying the entire codebase.

16. Scalability

The structure should allow new events, pages, or features to be added easily in the future without affecting existing sections.

17. Minimal Loading Time

The design should ensure lightweight pages with optimized images, minimal scripts, and efficient rendering to reduce load time.

18. Cross-Platform Support

The website should work not only on browsers but also on different platforms like Windows, macOS, Android, and iOS.

19. Professional Aesthetic

The color scheme, layout, and typography should reflect a modern and high-quality appearance suitable for an institutional fest website.

20. Intuitive Content Organization

Information should be arranged in a logical flow—Home → Events → Schedule → Gallery → Registration → Contact—so users can navigate naturally.

CHAPTER 5

IMPLEMENTATION

The implementation of the College Fest Website follows a structured approach, combining HTML, CSS, and JavaScript to create a complete, interactive web application. The process is carried out in multiple stages to ensure clarity, performance, and user satisfaction.

1. HTML Structure Development

- The website layout is built using semantic HTML5 tags such as `<header>`, `<nav>`, `<section>`, `<main>`, `<aside>`, and `<footer>`.
- Unique IDs (e.g., `#eventsGrid`, `#scheduleBody`, `#galleryGrid`) are assigned to elements so JavaScript can dynamically insert content.
- Form elements such as `<input>`, `<textarea>`, and `<select>` are implemented with validation-friendly attributes like `required`.
- Navigation links are anchored with `href="#section-id"` to allow smooth section scrolling.

2. CSS Styling and Page Layout

- A global theme is created using CSS variables (`--accent`, `--bg`, `--card`) for easy color customization.
- Flexbox and CSS Grid are used to create responsive layouts for the hero section, events grid, schedule table, and gallery.
- Media queries ensure the site adapts to mobile screen sizes by adjusting grid columns and hiding desktop-only navigation.
- Buttons, cards, images, and forms are styled for consistency and visual appeal, improving user experience.

3. JavaScript-Based Dynamic Rendering

- Arrays (`EVENTS`, `SCHEDULE`, `GALLERY`) hold all dynamic content, making the website scalable and easy to update.
- Functions like `renderEvents()`, `renderSchedule()`, and `renderGallery()` generate HTML dynamically using `innerHTML`.
- Event filtering is implemented through condition-based rendering triggered by category buttons (e.g., music, tech, sports).
- The registration form uses `event.preventDefault()` for custom validation and feedback messages.

4. Countdown Timer Implementation

- A JavaScript timer is created using setInterval() to update days, hours, minutes, and seconds until the fest date.
- The timer calculates time differences using the Date() object and updates DOM elements every second.

5. Modal (Lightbox) Functionality

- Clicking gallery or event images triggers openModalImage() to display a large version in a modal box.
- The modal uses classList.add() and classList.remove() to toggle visibility.
- An “ESC key to close modal” feature improves accessibility and user control.

6. Form Handling and Local Storage

- Registration data is stored in browser localStorage as a demo, allowing users to submit multiple entries without backend integration.
- JSON.stringify() and JSON.parse() convert data for smooth storage and retrieval.
- The contact form also displays a message upon successful submission.

7. Smooth Scrolling Navigation

- The website uses scrollIntoView({ behavior: 'smooth' }) to transition between sections, improving flow and readability.
- All navbar links listen for click events to trigger smooth scrolling instead of default jump navigation.

8. Performance Optimization

- Images are loaded using loading="lazy" to improve performance on slower networks.
- External content is kept minimal to ensure fast loading and efficient performance.

9. Responsive & Cross-Device Testing

- The website is tested on different devices and screen sizes to ensure consistent user experience.
- Layout breakpoints maintain readability and usability on both small and large screens.

10. Code Organization & Maintainability

- HTML, CSS, and JavaScript are written in separate files (index.html, style.css, script.js) for cleaner maintenance.
- Functions are modular and reusable, making future updates simple and efficient.
- Variables and constants are grouped logically for readability.

CHAPTER 6

RESULTS WEBSITE SNAPSHOTS

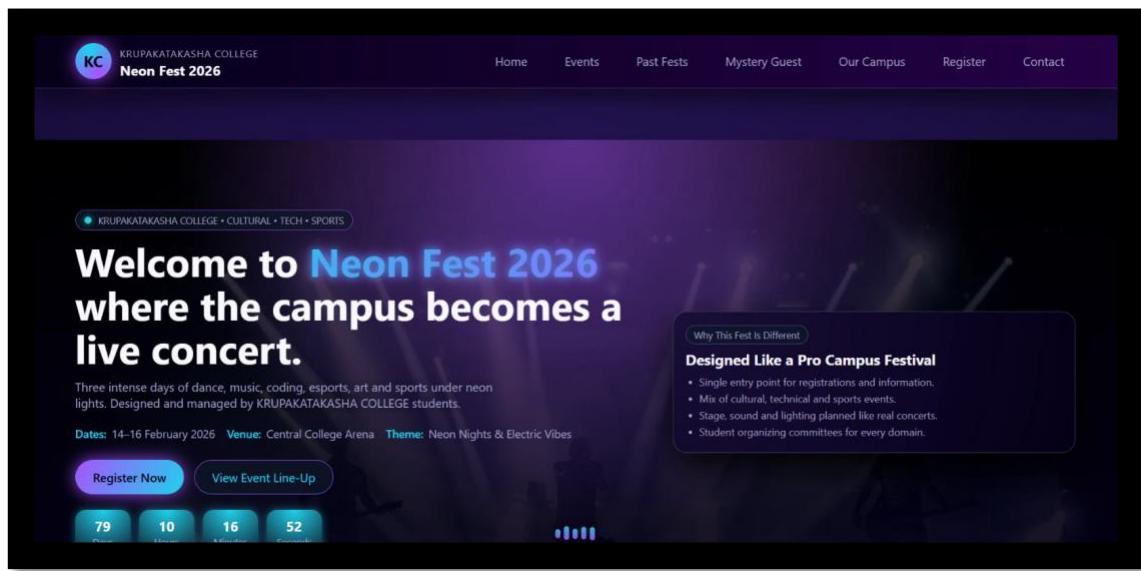


Fig.NO.6.1 HOME PAGE

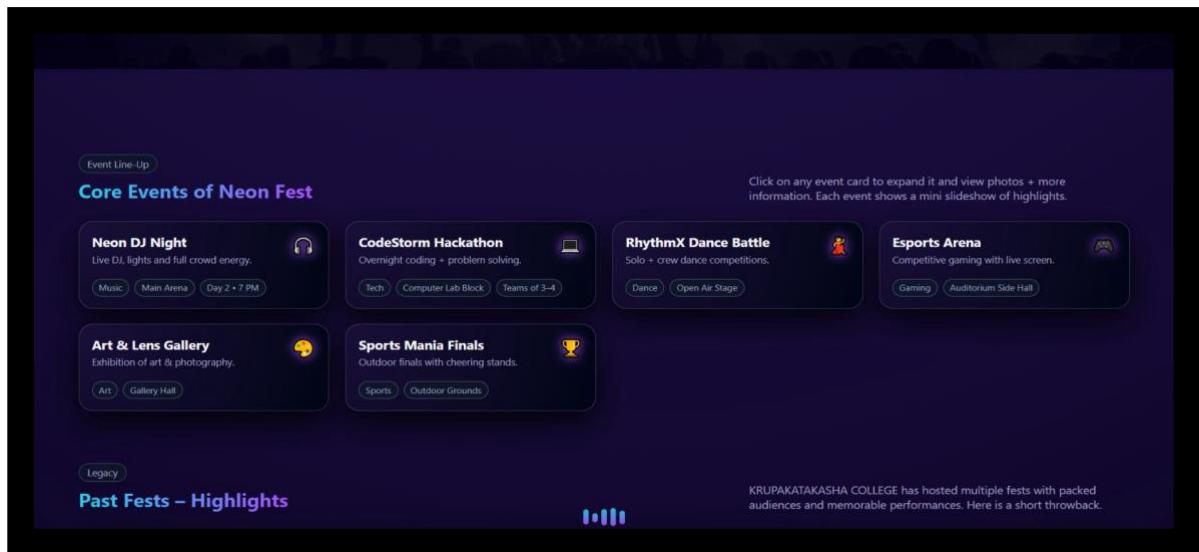


Fig.NO.6.2 EVENT LINE UP

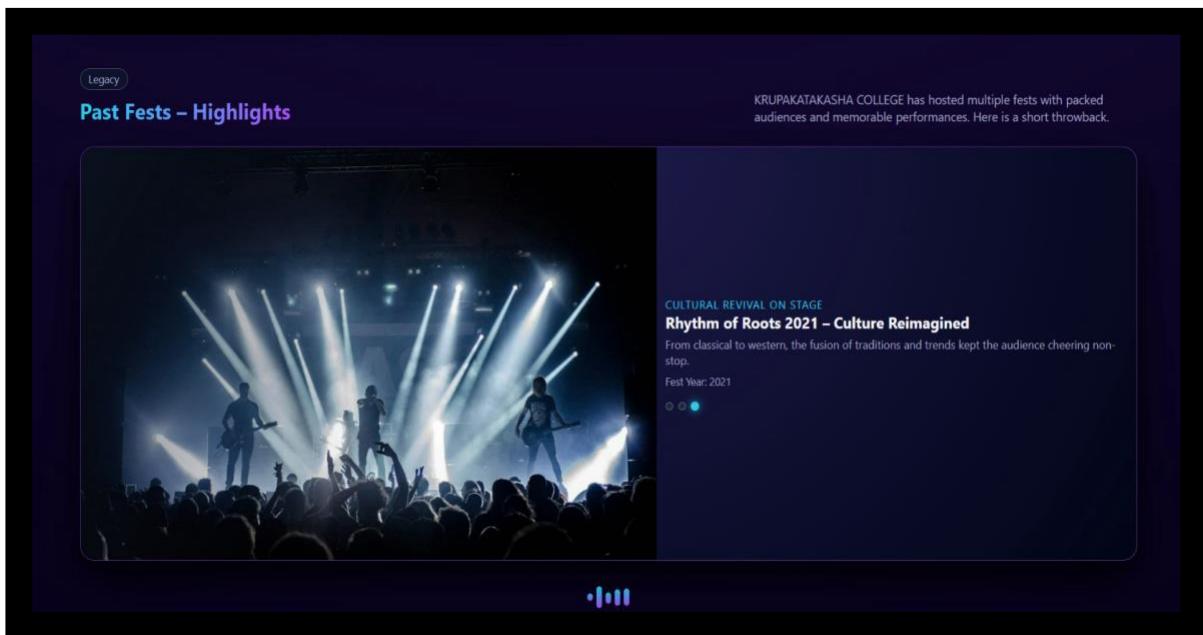


Fig.NO.6.3 PAST EVENTS HIGHLIGHTS

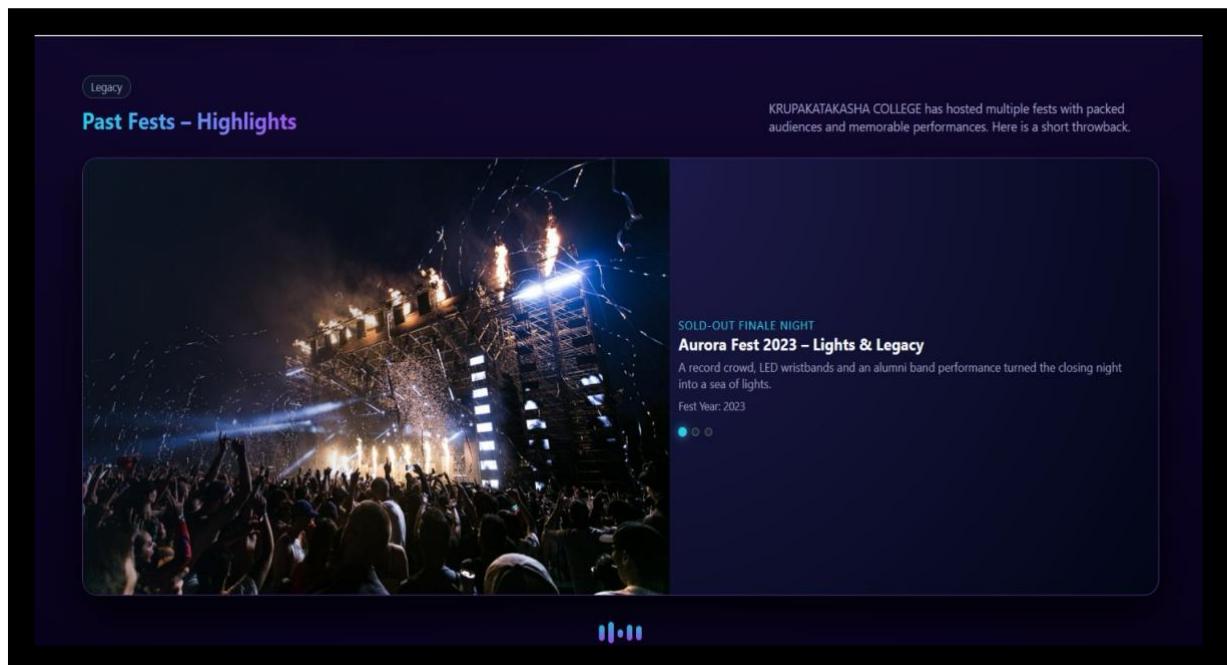


Fig.NO.6.4.PAST EVENTS HIGHLIGHTS SLIDE2

College fest website

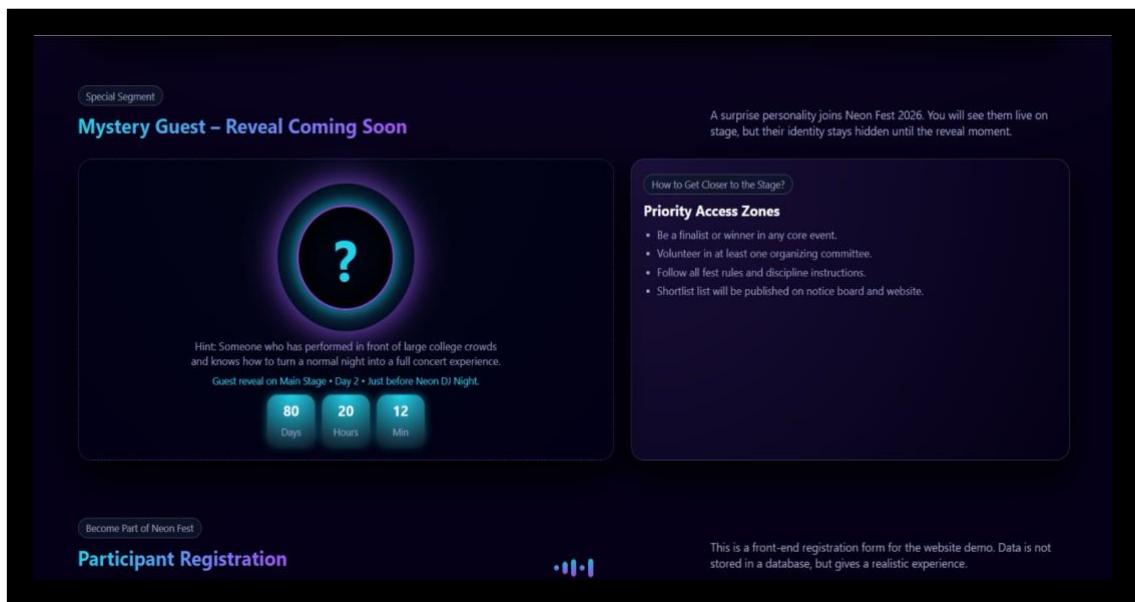


Fig.NO.6.5.MYSTERY GUEST REVEAL

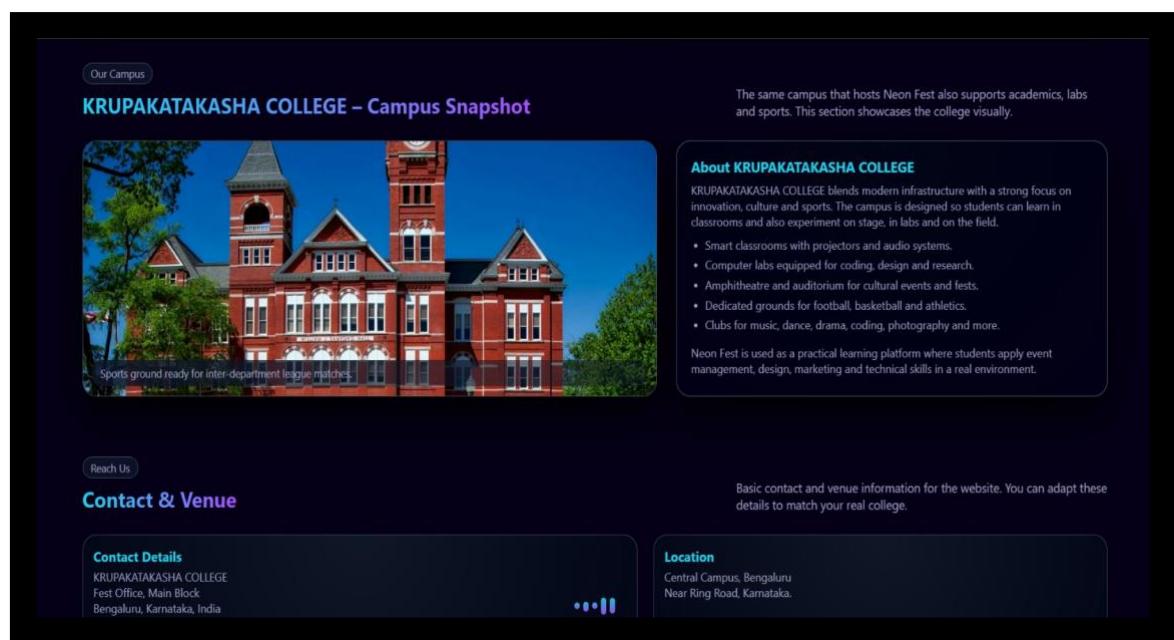


Fig.NO.6.6.CAMPUS SNAPSHOT

CHAPTER 7

CONCLUSION

The implementation of the College Fest Website successfully demonstrates how HTML, CSS, and JavaScript can be combined to build a fully functional, interactive, and visually appealing web application. The project fulfills its main objectives by providing a platform that showcases event details, schedules, gallery images, and registration features in an organized and user-friendly manner. Through responsive design techniques, the website adapts efficiently to different screen sizes, ensuring accessibility across mobile phones, tablets, and desktops.

The integration of JavaScript enhances user engagement by enabling dynamic features such as countdown timers, interactive event filtering, modal image viewing, and real-time form validation. The separation of design, structure, and interactivity into CSS, HTML, and JavaScript files ensures clean code organization and easier future maintenance.

Overall, the website achieves its goal of delivering an informative and engaging digital interface for a college fest, supporting both participants and organizers. This project also demonstrates essential web development concepts and serves as a strong foundation for future enhancements such as backend integration, automated registrations, and extended event management features.

REFERENCES

1. Mozilla Developer Network (MDN Web Docs)
<https://developer.mozilla.org/en-US>
2. W3Schools – Web Development Tutorials
<https://www.w3schools.com>
3. Freepik – Free Graphic Resources
<https://www.freepik.com/>
4. Unsplash – Free High-Quality Images
<https://unsplash.com/>
5. CSS-Tricks – CSS Guides & Techniques
<https://css-tricks.com/>
6. Stack Overflow – Programming Q&A Community
<https://stackoverflow.com/>
7. Google Fonts Documentation
<https://fonts.google.com/>
8. ChatGPT – OpenAI (Used for guidance, explanations, and documentation)
<https://chat.openai.com/>