

TITLE :- COLLEGE EVENT MANAGEMENT SYSTEM



INDEX:

List of figures/ tables	3
Abstract	4
Introduction	5
Study of the existing work/ Literature Review	7
Requirement specification –	8
a) What is the need for the project	
b) Flaws in the existing system,	
c) Advancements from the existing system	
d) Target audience	
Methodology	9
a) Model	
b) Flow diagram of the work	
c) Technologies used	
i) Description of technology	
d) Software and Hardware Requirements	
Code snippets	14
Results	20
a) Screenshots	
Conclusion and future scope	28
References	29
a. Journals/Conferences/Blogs/Articles/web-sites	

LIST OF FIGURES/TABLES:

FIGURE NAME	PAGE NO
1. Flow chart	9
2. Loginpage.html	14
3. Eventpage.html	15
4. Gallerypage.html	16
5. Coordinator-eventpage.html	17
6. Coordinator-Addeventpage.html	18
7. Adminpage.html	19
8. server.js	20
9. Login page	22
10. Admin Login page	22
11. Home Page	23
12. Event Page	24
13. Student Register Page	24
14. Gallerypage	25
15. About Us page	25
16. Coordinator Event Page	26
17. Coordinator Add Event Page	26
18. Admin Page	27

ABSTRACT:

This college event management system is a web-based platform designed to streamline event coordination within academic institutions. It enables students to register for events, automatically sending email confirmations upon successful registration. Event coordinators can efficiently add, update, and manage upcoming events, while the admin role oversees the application by managing coordinators and monitoring overall system functionality. The platform enhances engagement, simplifies event administration, and improves communication among students, coordinators, and administrators

INTRODUCTION:

Organizing events in a college setting involves managing multiple participants, schedules, and tasks, often leading to logistical challenges. Our college event management system aims to address these issues by providing a centralized, user-friendly platform for event coordination. Through this system, students can easily register for events and receive immediate confirmation via email, ensuring seamless participation. Event coordinators have the flexibility to create and manage events directly, enabling dynamic updates to event schedules and details. The admin role enhances system reliability by overseeing coordinators and maintaining system standards, thereby fostering a smooth and engaging event experience for all users. This platform not only simplifies event management but also supports higher student involvement and efficient organizational practices within the college community.. This College Event Management System is a web-based application designed to manage events for students, faculty, and administration. The system allows users to view, register, and manage upcoming and past events. It uses a frontend built with HTML, CSS, and JavaScript, while the backend is developed with Node.js and a MySQL database to store event data, user information, and registrations.

OBJECTIVES :

Here are some key objectives for a college event management system:

- 1. User-Friendly Interface:** Develop a clean and intuitive interface for both administrators and users to navigate easily and efficiently manage events.
- 2. Event Creation and Management:** Enable users to create, edit, and delete events with detailed information such as date, time, location, description, and images.
- 3. Event Registration:** Implement a registration system that allows users to sign up for events, including automated confirmation emails and reminders.
- 4. Event Categorization and Filtering:** Allow events to be categorized and filtered by type, date, or location to help users find relevant events quickly.
- 5. Past and Upcoming Events Display:** Provide clear separation and display of upcoming and past events, with the ability to view details and summaries of both.
- 6. Automatic Notifications and Alerts:** Integrate notifications and alerts for users and administrators regarding upcoming events, registration deadlines, and changes.

KEY FEATURES:

Here are some key features for your college event management system:

Student Registration: Students can easily register for upcoming events with automated email confirmations sent upon successful registration.

Event Creation and Management: Coordinators have access to add, update, and manage events, keeping the information dynamic and current.

Admin Oversight: An admin manages coordinator accounts, monitors event listings, and ensures system integrity.

Automated Notifications: Email notifications for event confirmations, reminders, and updates enhance communication.

User-Friendly Interface: Simple, accessible design for seamless navigation across student, coordinator, and admin roles.

Event Tracking: Allows for tracking student participation and maintaining a history of past events.

Study of Existing Work/Literature Review:

1. **EVENT MANAGEMENT SYSTEM (EMS):** Authored by J.R.V. Jeny; P. Sadhana; B. Jeevan Kumar; S. Leela Abhishek; T. Sai Chander a Traditional event management involved manual processes with limited technology integration. The proposed system enhances event coordination by integrating social media for promotion, remote access for event managers, and customizable event logistics, making it more user-friendly and dynamic than earlier systems.
2. **COLLEGE EVENT MANAGEMENT SYSTEM:** This paper is authored by Gaurav Thombare , Pavan Jadhav, Sachin Dhere, Prasad Jadhav, Dr. K.N. Honwadkar .Older college event systems faced communication and coordination challenges. The proposed CEMS improves efficiency by using a digital platform for better collaboration, communication, and user feedback, offering a significant upgrade over traditional methods.
3. **Modern Technological Solutions for Event Management:** This paper is authored by Gaurav Thombare, Pavan Jadhav , Sachin Dhere , Prasad Jadhav , Priyanka Raikar .It highlights the transition from manual processes to digital platforms using technologies like Next.js, React.js, and MongoDB, which enable better communication and coordination in event management, overcoming the limitations of older systems.
4. **College Event Organizer:**Authored by **Nirupama B.K.** This system simplifies event approval and notifications by automating communication between students and administrators. This addresses issues found in earlier manual processes, making event management faster and more accessible.
5. **Centrally Event Management System:** Authors **Harika T., Siva Ranjani P., Mohana Kumari K., and Sushma Sri V** .This paper provided Older systems struggled with data fragmentation and inefficient communication. The proposed system centralizes event information, improves access, and introduces features like digital attendance tracking, offering a more organized and user-friendly experience.

Across the existing literature, the move from traditional, manual event management methods to sophisticated, digital platforms is a consistent theme. Each system introduces features aimed at improving user engagement, communication, and efficiency—whether through social media integration, calendar synchronization, or real-time data access. By addressing common challenges like inefficient communication, approval delays, and lack of data centralization, these systems represent a significant leap forward from earlier event management practices. The findings from the reviewed works underscore the value of modernized, technology-driven solutions in optimizing event management processes, particularly in academic environments.

Improvements Over Existing Systems:

- Automated event registration and management.
- User-friendly and responsive design.
- Integrated image galleries showcasing past events.

Requirement Specification :

a. What is the need for the project?

Managing college events manually is time-consuming and error-prone. This system automates event creation, registration, and tracking, providing an efficient platform for both event organizers and participants.

b. Flaws in the existing system

- Manual tracking of registrations.
- Difficulty in disseminating information about events.
- Lack of integrated gallery and history features for past events

c. Advancements from the existing system

- User-friendly registration form with validation.
- Image galleries for past event highlights.
- Easy event filtering
- Categorised events (Technical, Non - technical, Cultural, Workshops, Sports).

d. Target Audience

The system is designed for college students, faculty, and administrators to manage and participate in college events.

Methodology :

a. Model :

The system follows a client-server model. The frontend is built with HTML, CSS, and JavaScript, while the backend uses Node.js and MySQL.

b. Flow Diagram of the Work:



FIG 1: Flowchart

c. Technologies Used:

- **Frontend:** HTML, CSS, JavaScript
- **Backend:** Node.js (Express.js)
- **Database:** MySQL

i) Description of technologies :

Frontend Technologies:



HTML:

HTML (HyperText Markup Language) is the standard markup language used to create and structure content on the web. It provides the fundamental building blocks of web pages, including elements for text, links, images, and multimedia.

CSS:

CSS (Cascading Style Sheets) is the styling language used to define the look and layout of web pages. CSS allows for detailed control over the presentation of HTML elements, including color, typography, spacing, and positioning. It introduces advanced features such as responsive design techniques, animations, transitions, and custom fonts, which enhance the visual appeal and usability of web pages.

Javascript:

JavaScript is a high-level, dynamic programming language that enables interactive and dynamic features on web pages. It runs in the browser and allows developers to create responsive user interfaces, validate forms, handle events, and manipulate the Document Object Model (DOM). JavaScript is essential for adding client-side logic and interactivity, such as updating content without refreshing the page and implementing animations.

Backend Technologies:



Node.js: Node.js is a JavaScript runtime built on Chrome's V8 JavaScript engine. It enables server-side execution of JavaScript code, allowing developers to build scalable and high-performance web applications. Node.js uses an event-driven, non-blocking I/O model, which makes it suitable for handling a large number of simultaneous connections efficiently. It is commonly used for building APIs and backend services.

Database Technology:



MySQL: MySQL is a popular open-source relational database management system (RDBMS) that uses Structured Query Language (SQL) for managing and querying data. It is widely used for its reliability, ease of use, and performance. MySQL organizes data into tables with predefined schemas, supports complex queries and transactions, and ensures data integrity through features like foreign keys and indexing. It is commonly used for storing and managing data in web applications, including user information and event details.

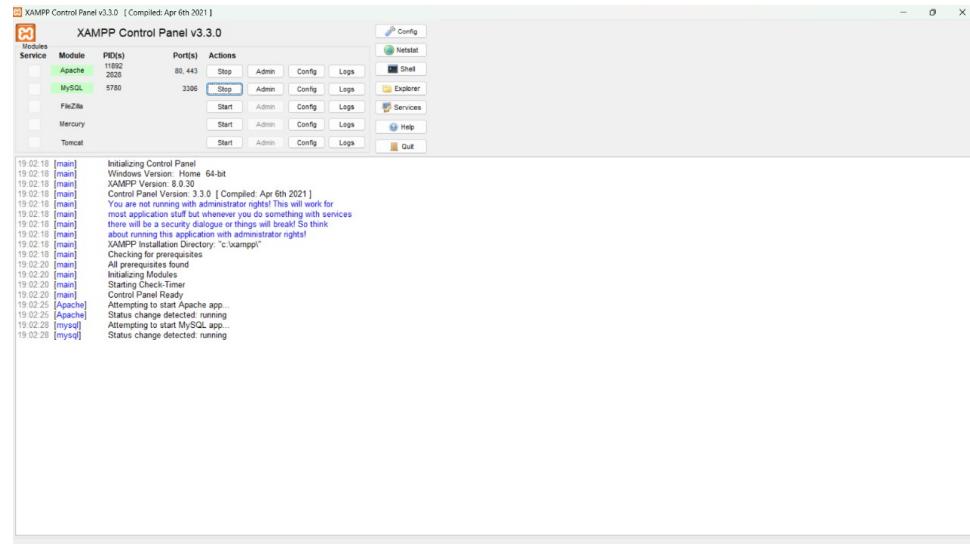
d. Software and Hardware requirements:

Software:

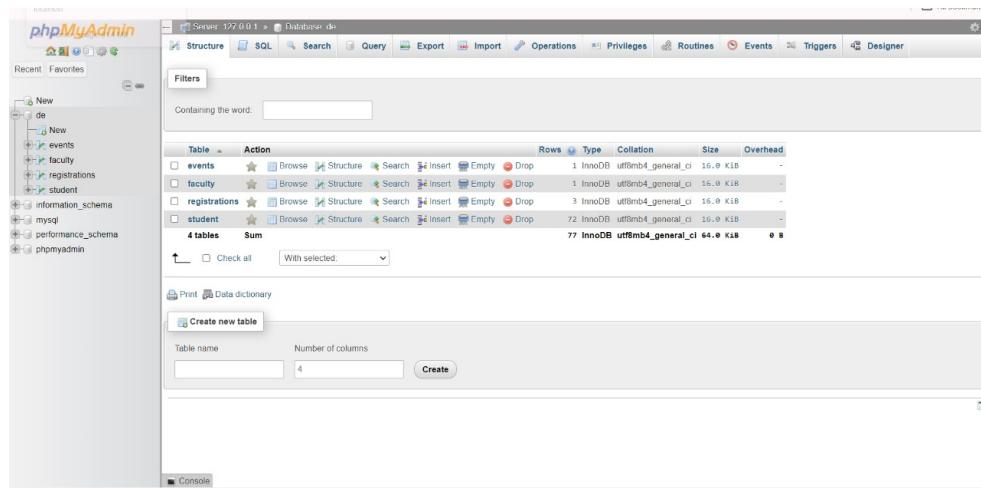
- Node.js (v14+)
- MySQL (v8.0+)
- Web Browser (Chrome/Firefox)

HOW TO RUN THIS PROJECT:

1. Start the MYSQL and APACHE server and go to MYSQL admin to add database.



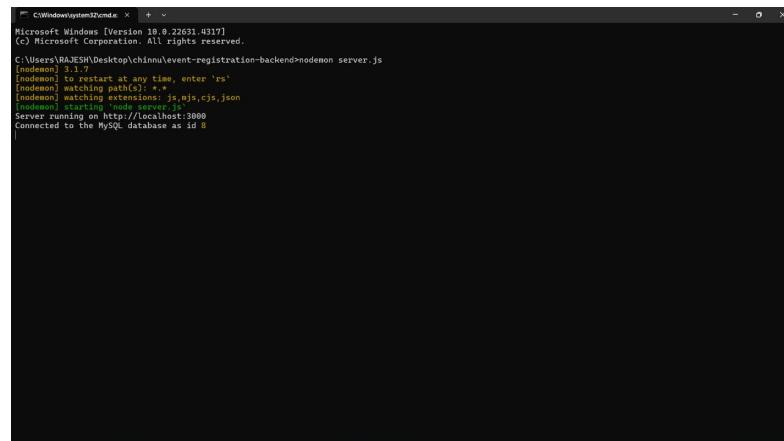
2. import the database into phpMyAdmin databases.



3. install all the necessary modules for running node.js in your command prompt.

- Sudo apt install nodejs
- Sudo apt install npm
- Npm install nodemailer

4. In command prompt give a command called “nodemon sever.js” to run the application in localhost.



```
C:\Windows\system32\cmd.exe + ~ Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\RAJESH\Desktop\chnml\event-registration-backend>nodemon server.js
[nodemon] 3.1.7
[nodemon] to restart at any time, enter 'rs'
[nodemon] watching path(s): *
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting 'node server.js'
Server running on http://localhost:3000
Connected to the MySQL database as id 8
'
```

5. It provides a localhost ip address. copy and paste it in the browser.



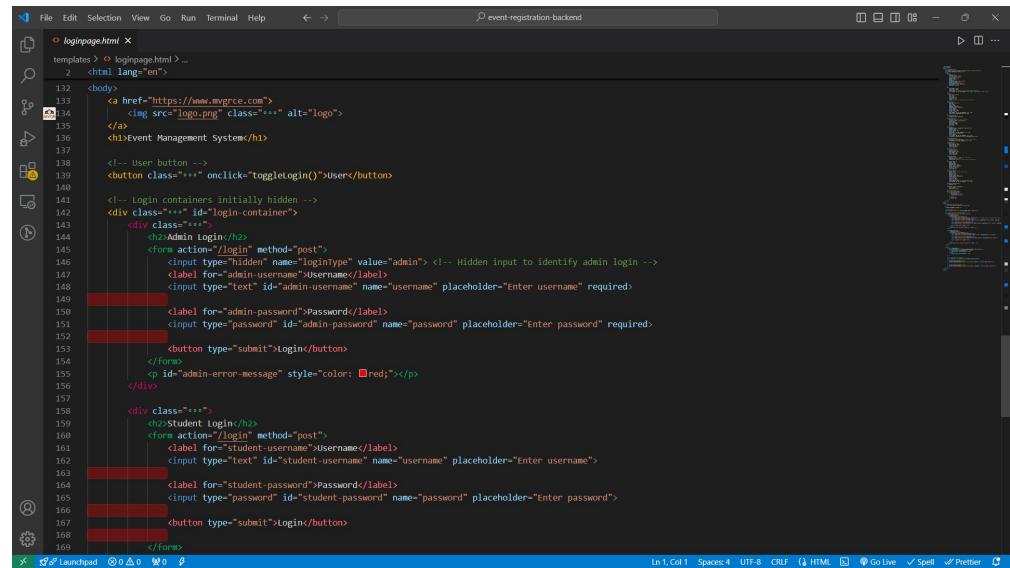
6. The application login page will be displayed.



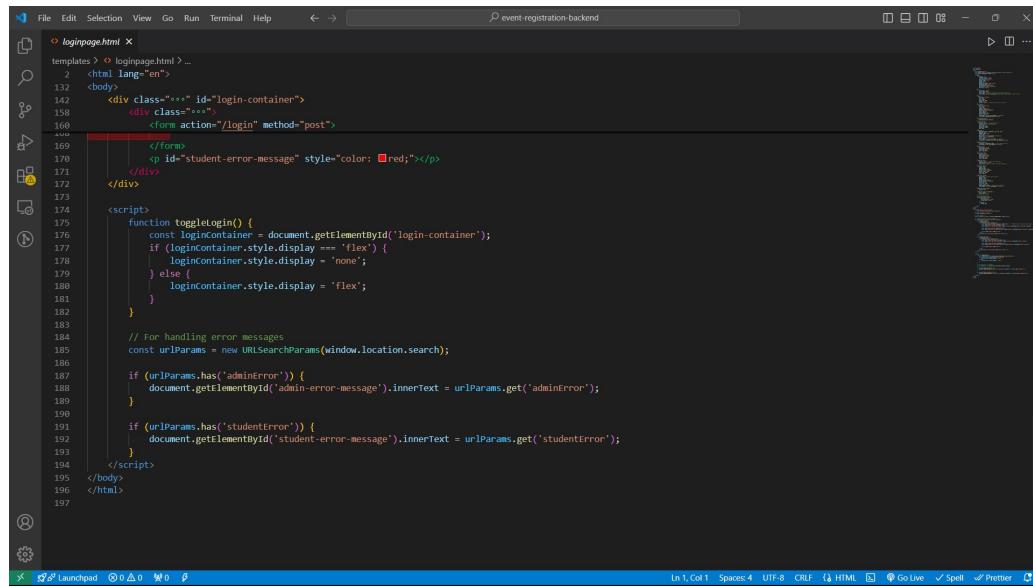
Code Snippets:

Html codes:

Loginpage.html:



```
<html lang="en">
  <body>
    <a href="https://www.myrcce.com">
      
    </a>
    <h1>Event Management System</h1>
    <!-- User button -->
    <button class="user" onclick="toggleLogin()>User</button>
    <!-- Login containers initially hidden -->
    <div class="" id="login-container">
      <div class="">
        <h2>Admin Login</h2>
        <form action="/login" method="post">
          <input type="hidden" name="logintype" value="admin" /> <!-- Hidden input to identify admin login -->
          <label for="admin-username">Username</label>
          <input type="text" id="admin-username" name="username" placeholder="Enter username" required>
          <br/>
          <label for="admin-password">Password</label>
          <input type="password" id="admin-password" name="password" placeholder="Enter password" required>
          <br/>
          <button type="submit">Login</button>
        </form>
        <p id="admin-error-message" style="color: red;"></p>
      </div>
    </div>
    <div class="">
      <h2>Student Login</h2>
      <form action="/login" method="post">
        <label for="student-username">Username</label>
        <input type="text" id="student-username" name="username" placeholder="Enter username" required>
        <br/>
        <label for="student-password">Password</label>
        <input type="password" id="student-password" name="password" placeholder="Enter password" required>
        <br/>
        <button type="submit">Login</button>
      </form>
      <p id="student-error-message" style="color: red;"></p>
    </div>
  </body>
</html>
```



```
<html lang="en">
  <body>
    <div class="" id="login-container">
      <div class="">
        <form action="/login" method="post">
          <br/>
          <div>
            <script>
              function toggleLogin() {
                const loginContainer = document.getElementById('login-container');
                if (loginContainer.style.display === 'flex') {
                  loginContainer.style.display = 'none';
                } else {
                  loginContainer.style.display = 'flex';
                }
              }
            </script>
            // For handling error messages
            const urlParams = new URLSearchParams(window.location.search);
            if (urlParams.has('adminError')) {
              document.getElementById('admin-error-message').innerText = urlParams.get('adminError');
            }
            if (urlParams.has('studentError')) {
              document.getElementById('student-error-message').innerText = urlParams.get('studentError');
            }
          </div>
        </form>
        <p id="student-error-message" style="color: red;"></p>
      </div>
    </div>
  </body>
</html>
```

Fig 2: Loginpage.html

Eventpage.html:

```

File Edit Selection View Go Run Terminal Help < - > event registration-backend
templates > events.html > html > head > style > event-test.css
171 <html lang="en">
172   <head>
173     <!-- Header Section -->
174     <header>
175       <h1>Events</h1>
176     </header>
177     <!-- Main Content -->
178     <div class="main">
179       <!-- Upcoming Events Section -->
180       <div class="upcoming">
181         <h2>Upcoming Events</h2>
182         <!-- Technical Events -->
183         <button class="technical">Technical</button>
184         <div class="tech">
185           <!-- Dynamic content will be inserted here -->
186         </div>
187         <!-- Non-technical Events -->
188         <button class="non-technical">Non-technical</button>
189         <div class="non-tech">
190           <!-- Dynamic content will be inserted here -->
191         </div>
192         <!-- Cultural Events -->
193         <button class="cultural">Cultural</button>
194         <div class="cultural">
195           <!-- Dynamic content will be inserted here -->
196         </div>
197         <!-- Sports Events -->
198         <button class="sports">Sports</button>
199         <div class="sports">
200           <!-- Dynamic content will be inserted here -->
201         </div>
202       <!-- Backdrops -->
203       
204       
205       
206     </div>
207   </body>
208 </html>

```



```

File Edit Selection View Go Run Terminal Help < - > event-registration-backend
templates > events.html > html > head > style > event-test.css
171 <html lang="en">
172   <head>
173     <!-- Header Section -->
174     <header>
175       <h1>Events</h1>
176     </header>
177     <!-- Main Content -->
178     <div class="main">
179       <!-- Past Events Section -->
180       <div class="past">
181         <h2>Past Events</h2>
182         <div class="past-item">
183           
184           <div class="desc">
185             
186             <div class="desc">
187               
188             <!-- Add more past event images as needed -->
189           </div>
190         </div>
191       <!-- Backdrops -->
192       
193       
194       
195     </div>
196   </body>
197 </html>

```



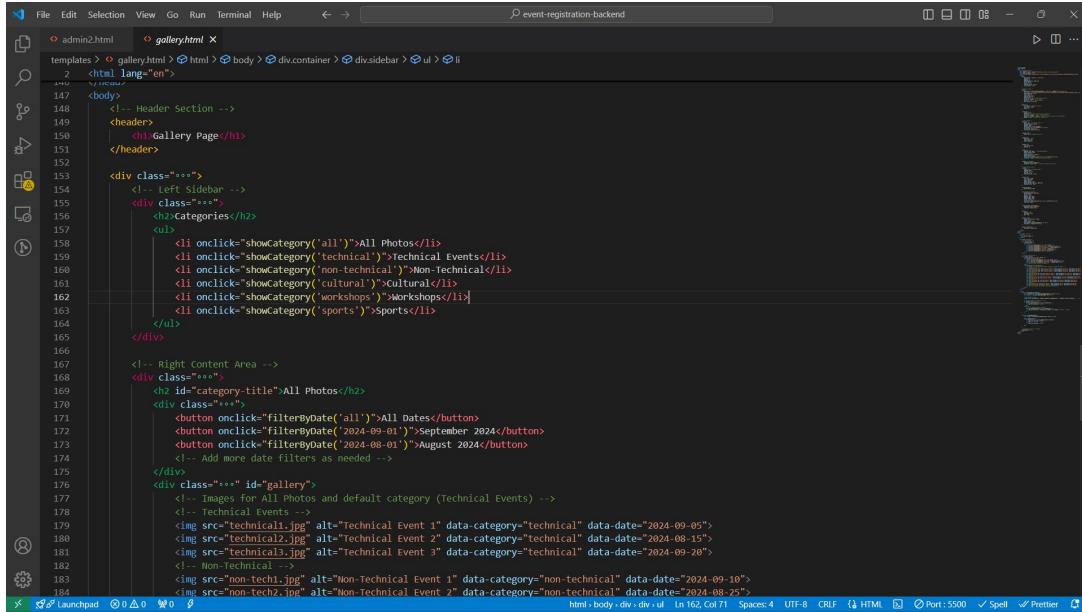
```

File Edit Selection View Go Run Terminal Help < - > event-registration-backend
templates > events.html > html > head > style > event-test.css
171 <html lang="en">
172   <head>
173     <script>
174       .then(events => {
175         // Sort events by category and populate the respective sections
176         events.forEach(event => {
177           const eventItem = document.createElement('div');
178           eventItem.innerHTML =
179             `
180             <div class="info">
181               <strong>${event.name}</strong>
182               <strong>${event.description}</strong>
183               <strong>${event.date}</strong>
184               <strong>${event.location}</strong>
185               <strong>${event.starttime}</strong>
186               <strong>${event.slots}</strong>
187               <strong>${event.slots}</strong>
188             </div>`;
189           document.querySelector(`#${event.category}`).innerHTML += eventItem;
190         });
191       });
192     </script>
193     .catch(error => {
194       console.error('Error fetching events:', error);
195     });
196   </head>
197   <body>
198     <!-- Header Section -->
199     <header>
200       <h1>Events</h1>
201     </header>
202     <!-- Main Content -->
203     <div class="main">
204       <!-- Technical Events -->
205       <div class="technical">
206         <button class="technical">Technical</button>
207         <div class="tech">
208           <!-- Dynamic content will be inserted here -->
209         </div>
210       <!-- Non-technical Events -->
211       <div class="non-technical">
212         <button class="non-technical">Non-technical</button>
213         <div class="non-tech">
214           <!-- Dynamic content will be inserted here -->
215         </div>
216       <!-- Cultural Events -->
217       <div class="cultural">
218         <button class="cultural">Cultural</button>
219         <div class="cultural">
220           <!-- Dynamic content will be inserted here -->
221         </div>
222       <!-- Sports Events -->
223       <div class="sports">
224         <button class="sports">Sports</button>
225         <div class="sports">
226           <!-- Dynamic content will be inserted here -->
227         </div>
228       <!-- Backdrops -->
229       
230       
231       
232     </div>
233   </body>
234 </html>

```

Fig 3: Eventpage.html

Gallerypage.html:

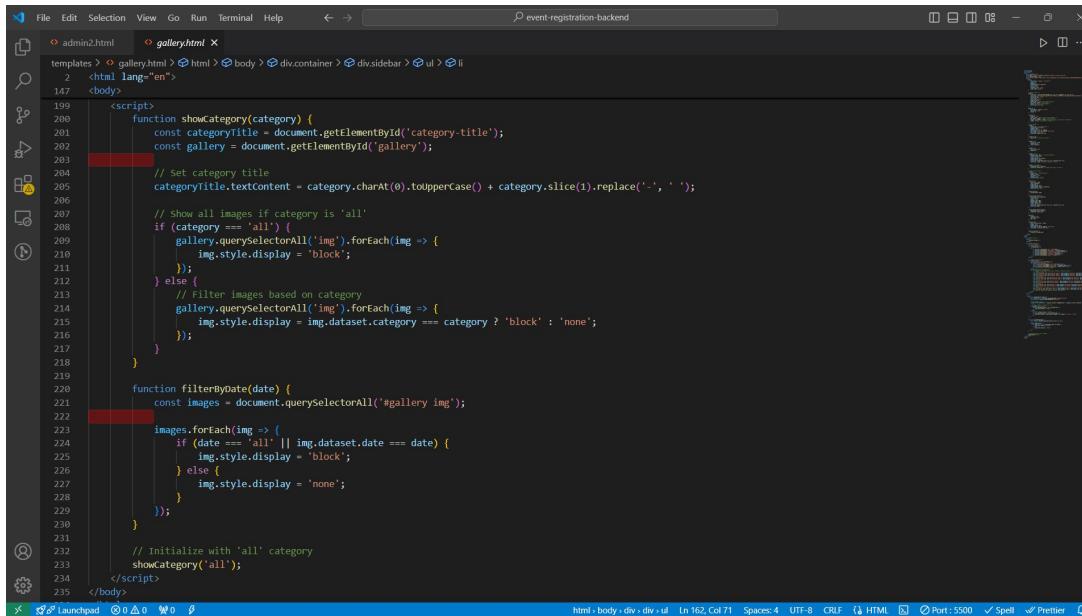


The screenshot shows the Visual Studio Code interface with the file `gallerypage.html` open. The code is an HTML template for a gallery page. It includes a sidebar with categories like All Photos, Technical Events, Non-Technical, Cultural, Workshops, and Sports. Below the sidebar is a section for date filters with buttons for All Dates, September 2024, and August 2024. The main content area contains images categorized by event type, such as technical and non-technical events.

```
<html lang="en">
<body>
    <!-- Header Section -->
    <header>
        <h1>Gallery Page</h1>
    </header>

    <div class="***">
        <!-- Left Sidebar -->
        <div class="***">
            <h2>Categories</h2>
            <ul>
                <li onclick="showCategory('all')>All Photos</li>
                <li onclick="showCategory('technical')>Technical Events</li>
                <li onclick="showCategory('non-technical')>Non-Technical</li>
                <li onclick="showCategory('cultural')>Cultural</li>
                <li onclick="showCategory('workshops')>Workshops</li>
                <li onclick="showCategory('sports')>Sports</li>
            </ul>
        </div>

        <!-- Right Content Area -->
        <div class="***">
            <h2 id="category-title">All Photos</h2>
            <div class="***">
                <button onclick="filterByDate('all')>All Dates</button>
                <button onclick="filterByDate('2024-09-01')>September 2024</button>
                <button onclick="filterByDate('2024-08-01')>August 2024</button>
                <!-- Add more date filters as needed -->
            </div>
            <div class="***" id="gallery">
                <!-- Images for All Photos and default category (Technical Events) -->
                <!-- Technical Events -->
                
                
                
                <!-- Non-Technical -->
                
                
            </div>
        </div>
    </div>
</body>
```



The screenshot shows the Visual Studio Code interface with the file `gallerypage.html` open, focusing on the script block. The code defines two functions: `showCategory` and `filterByDate`. The `showCategory` function iterates over all images and sets their display style based on the selected category ('all' or a specific category like 'technical'). The `filterByDate` function iterates over all images and hides them if their dataset date does not match the selected filter ('all' or a specific date like '2024-09-01').

```
<script>
    function showCategory(category) {
        const categoryTitle = document.getElementById('category-title');
        const gallery = document.getElementById('gallery');

        // Set category title
        categoryTitle.textContent = category.charAt(0).toUpperCase() + category.slice(1).replace('-', ' ');

        // Show all images if category is 'all'
        if (category === 'all') {
            gallery.querySelectorAll('img').forEach(img => {
                img.style.display = 'block';
            });
        } else {
            // Filter images based on category
            gallery.querySelectorAll('img').forEach(img => {
                img.style.display = img.dataset.category === category ? 'block' : 'none';
            });
        }
    }

    function filterByDate(date) {
        const images = document.querySelectorAll('#gallery img');

        images.forEach(img => {
            if (date === 'all' || img.dataset.date === date) {
                img.style.display = 'block';
            } else {
                img.style.display = 'none';
            }
        });
    }
</script>
</body>
```

Fig 4:Gallerypage.html

Coordinator-eventpage.html:

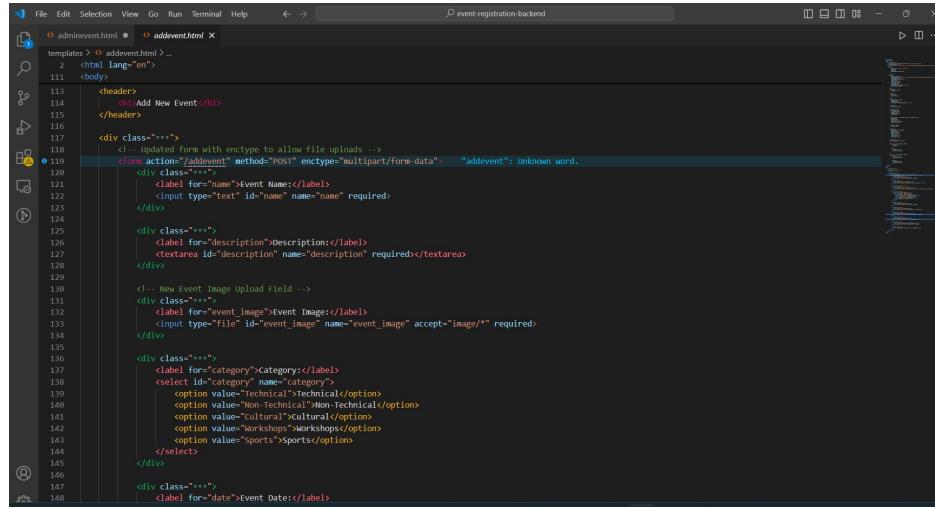
```

File Edit Selection View Go Run Terminal Help <- > event-registration-backend
templates > o adminevent.html < ...
  2 <html lang="en">
  3   <body>
  4     <div class="header">
  5       <div>Events</div>
  6     </div>
  7     <div class="main">
  8       <div><!-- Past Events Section -->
  9         <div class="list-item">
 10           <div class="list-item">
 11             <div id="past-events-thread">
 12               <div><!-- Past Events -->
 13                 <div class="list-item">
 14                   <div class="list-item">
 15                     
 16                     <div class="list-item">
 17                       <div class="list-item">
 18                         <p>Cyber Security</p>
 19                         <p>A brief description of what happened in Past Event 1. It was a great success!</p>
 20                     </div>
 21                   </div>
 22                 </div>
 23               </div>
 24             </div>
 25           </div>
 26         </div>
 27       </div>
 28       <div><!-- Upcoming Events Section -->
 29         <div class="list-item">
 30           <div class="list-item">
 31             <div><!-- Add New Event Button -->
 32               <button>Add Event</button>
 33             </div>
 34           </div>
 35         </div>
 36       </div>
 37     </div>
 38   </script>
 39   <script>
 40     function addEvent() {
 41       // simulate adding an event - you can replace this with actual functionality
 42       const newEvent = {
 43         title: "New Event",
 44         description: "This is a newly added event."
 45       };
 46       // Create a new event container element
 47       const eventContainer = document.createElement("div");
 48       eventContainer.classList.add("event-container");
 49       const eventImage = document.createElement("img");
 50       eventImage.src = newEvent.image;
 51       eventImage.alt = newEvent.title;
 52       const eventDetails = document.createElement("div");
 53       eventDetails.classList.add("event-details");
 54       const eventTitle = document.createElement("h3");
 55       eventTitle.classList.add("event-title");
 56       eventTitle.textContent = newEvent.title;
 57       const eventDescription = document.createElement("p");
 58       eventDescription.classList.add("event-description");
 59       eventDescription.textContent = newEvent.description;
 60       // Append the details to the container
 61       eventTitle.appendChild(eventImage);
 62       eventDetails.appendChild(eventTitle);
 63       eventDetails.appendChild(eventDescription);
 64       eventContainer.appendChild(eventImage);
 65       eventContainer.appendChild(eventDetails);
 66       // Append the new event to the past events thread
 67       document.getElementById("past-events-thread").appendChild(eventContainer);
 68       window.location.href = "adminevent.html";
 69     }
 70   // Function to fetch events and display them
 71 </script>
 72 </body>
 73 </html>

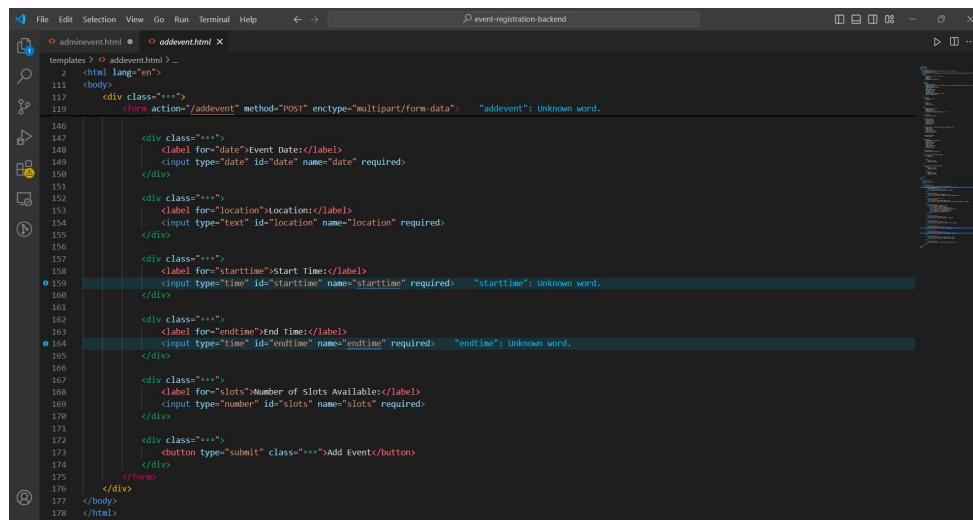
```

Fig 5: coordinator-eventpage.html

Cordinator-addevent.html:



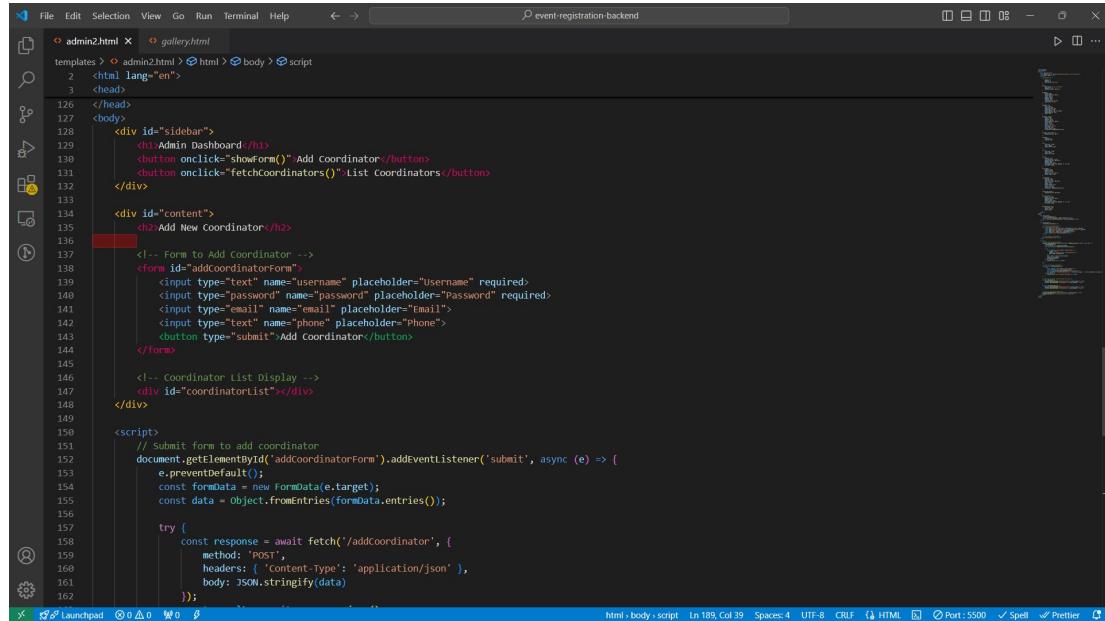
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <title>Add New Event</title>
</head>
<body>
    <div class="form">
        <form action="/addevent" method="POST" enctype="multipart/form-data">
            <div>
                <label for="name">Event Name:</label>
                <input type="text" id="name" name="name" required>
            </div>
            <div>
                <label for="description">Description:</label>
                <textarea id="description" name="description" required></textarea>
            </div>
            <div>
                <label for="event_image">Event Image:</label>
                <input type="file" id="event_image" name="event_image" accept="image/*" required>
            </div>
            <div>
                <label for="category">Category:</label>
                <select id="category" name="category">
                    <option value="technical">Technical</option>
                    <option value="Non-technical">Non-technical</option>
                    <option value="cultural">Cultural</option>
                    <option value="workshops">Workshops</option>
                    <option value="Sports">Sports</option>
                </select>
            </div>
            <div>
                <label for="date">Event Date:</label>
                <input type="date" id="date" name="date" required>
            </div>
            <div>
                <label for="starttime">Start Time:</label>
                <input type="time" id="starttime" name="starttime" required>
            </div>
            <div>
                <label for="location">Location:</label>
                <input type="text" id="location" name="location" required>
            </div>
            <div>
                <label for="endtime">End Time:</label>
                <input type="time" id="endtime" name="endtime" required>
            </div>
            <div>
                <label for="slots">Number of Slots Available:</label>
                <input type="number" id="slots" name="slots" required>
            </div>
            <div>
                <button type="submit" class="button">Add Event</button>
            </div>
        </form>
    </div>
</body>
</html>
```



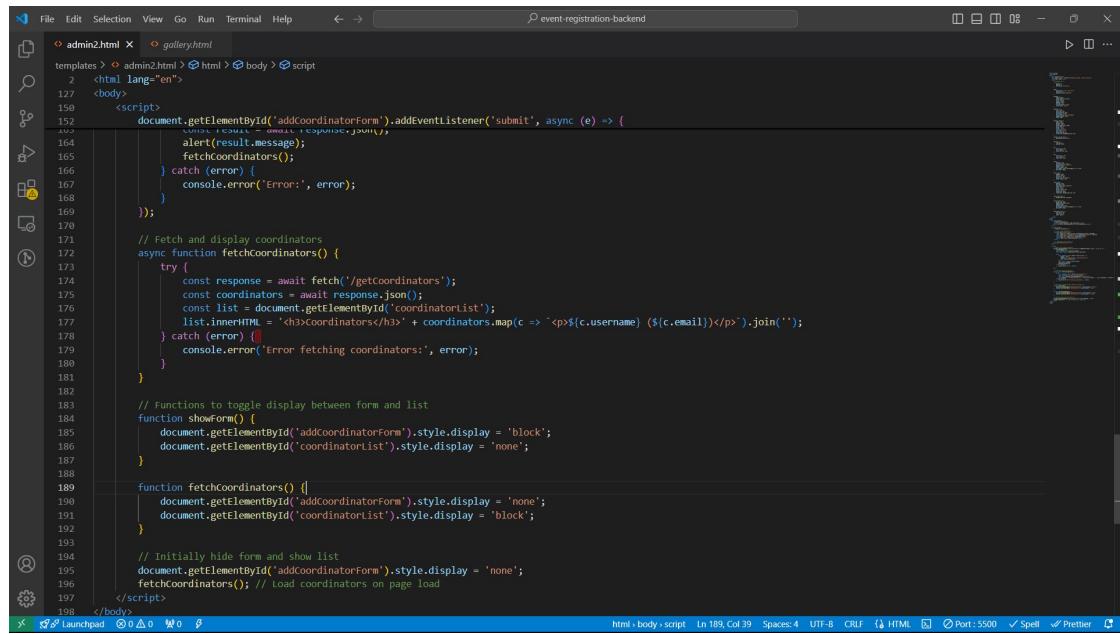
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <title>Add New Event</title>
</head>
<body>
    <div class="form">
        <form action="/addevent" method="POST" enctype="multipart/form-data">
            <div>
                <label for="name">Event Name:</label>
                <input type="text" id="name" name="name" required>
            </div>
            <div>
                <label for="description">Description:</label>
                <textarea id="description" name="description" required></textarea>
            </div>
            <div>
                <label for="event_image">Event Image:</label>
                <input type="file" id="event_image" name="event_image" accept="image/*" required>
            </div>
            <div>
                <label for="category">Category:</label>
                <select id="category" name="category">
                    <option value="technical">Technical</option>
                    <option value="Non-technical">Non-technical</option>
                    <option value="cultural">Cultural</option>
                    <option value="workshops">Workshops</option>
                    <option value="Sports">Sports</option>
                </select>
            </div>
            <div>
                <label for="date">Event Date:</label>
                <input type="date" id="date" name="date" required>
            </div>
            <div>
                <label for="starttime">Start Time:</label>
                <input type="time" id="starttime" name="starttime" required>
            </div>
            <div>
                <label for="location">Location:</label>
                <input type="text" id="location" name="location" required>
            </div>
            <div>
                <label for="endtime">End Time:</label>
                <input type="time" id="endtime" name="endtime" required>
            </div>
            <div>
                <label for="slots">Number of Slots Available:</label>
                <input type="number" id="slots" name="slots" required>
            </div>
            <div>
                <button type="submit" class="button">Add Event</button>
            </div>
        </form>
    </div>
</body>
</html>
```

Fig 6:coordinator-addeventpage.html

AdminPage.html:



```
<html lang="en">
  <head>
    <body>
      <div id="sidebar">
        <h1>Admin Dashboard</h1>
        <button onclick="showForm()">Add Coordinator</button>
        <button onclick="fetchCoordinators()">List Coordinators</button>
      </div>
      <div id="content">
        <h2>Add New Coordinator</h2>
        <!-- Form to Add Coordinator -->
        <form id="addCoordinatorForm">
          <input type="text" name="username" placeholder="Username" required>
          <input type="password" name="password" placeholder="Password" required>
          <input type="email" name="email" placeholder="Email">
          <input type="text" name="phone" placeholder="Phone">
          <button type="submit">Add Coordinator</button>
        </form>
        <!-- Coordinator List Display -->
        <div id="coordinatorList"></div>
      </div>
      <script>
        // submit form to add coordinator
        document.getElementById('addCoordinatorForm').addEventListener('submit', async (e) => {
          e.preventDefault();
          const formData = new FormData(e.target);
          const data = Object.fromEntries(formData.entries());
          try {
            const response = await fetch('/addCoordinator', {
              method: 'POST',
              headers: { 'Content-Type': 'application/json' },
              body: JSON.stringify(data)
            });
          }
        });
      </script>
    </body>
  </html>
```



```
<html lang="en">
  <head>
    <body>
      <script>
        document.getElementById('addCoordinatorForm').addEventListener('submit', async (e) => {
          e.preventDefault();
          const response = await fetch('/addCoordinator');
          if (!response.ok) {
            alert(result.message);
            fetchCoordinators();
          } catch (error) {
            console.error('Error:', error);
          }
        });

        // Fetch and display coordinators
        async function fetchCoordinators() {
          try {
            const response = await fetch('/getCoordinators');
            const coordinators = await response.json();
            const list = document.getElementById('coordinatorList');
            list.innerHTML = '<h3>Coordinators</h3>' + coordinators.map(c => `<p>${c.username} (${c.email})</p>`).join('');
          } catch (error) {
            console.error('Error fetching coordinators:', error);
          }
        }

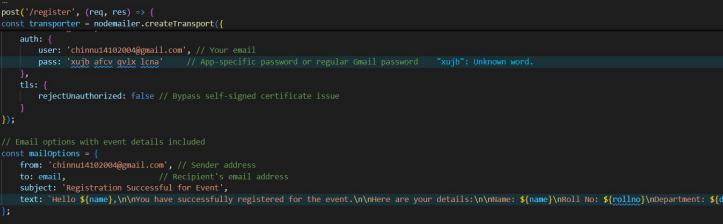
        // Functions to toggle display between form and list
        function showForm() {
          document.getElementById('addCoordinatorForm').style.display = 'block';
          document.getElementById('coordinatorList').style.display = 'none';
        }

        function fetchCoordinators() {
          document.getElementById('addCoordinatorForm').style.display = 'none';
          document.getElementById('coordinatorList').style.display = 'block';
        }

        // Initially hide form and show list
        document.getElementById('addCoordinatorForm').style.display = 'none';
        fetchCoordinators(); // load coordinators on page load
      </script>
    </body>
  </html>
```

Fig 7: Adminpage.html

Server.js:



```
File Edit Selection View Go Run Terminal Help < > event-registration-backend

adminevent.html JS server.js x

server.js
  app.post('/register', (req, res) => {
    const transporter = nodemailer.createTransport({
      auth: {
        user: 'chinmaya14102004@gmail.com', // Your email
        pass: 'xujh afcv gylx lcnh' // App-specific password or regular Gmail password "xujh": Unknown word.
      },
      tls: {
        rejectUnauthorized: false // Bypass self-signed certificate issue
      }
    });

    // Email options with event details included
    const mailOptions = {
      from: 'chinmaya14102004@gmail.com', // Sender's address
      to: mail, // Recipient's email address
      subject: 'Registration Successful for Event',
      text: `Hello $name,\nYou have successfully registered for the event.\n\nHere are your details:\nName: $name\nRoll No: ${rollno}\nDepartment: ${department}`
    };

    // Send email
    transporter.sendMail(mailOptions, (error, info) => {
      if (error) {
        console.log('Error occurred while sending email:', error);
        return res.status(500).send(`Error sending email: ${error.message}`);
      } else {
        console.log('Email sent: ' + info.response);
        res.status(200).send('Registration successful and email sent');
      }
    });
  });

  // Serve other static pages
  app.get('/about', (req, res) => {
    res.sendFile(path.join(__dirname, 'templates', 'about.html'));
  });
}

// Start the server
const port = process.env.PORT || 3001;
app.listen(port, () => {
  console.log(`Server is running on port ${port}`);
});
```

```
File Edit Selection View Go Run Terminal Help event-registration-backend
server.js > ...
  181 app.post('/addevent', (req, res) => { "addevent": Unknown word.
  182   upload(req, res, (err) => {
  183     const query = "INSERT INTO events (name, description, category, date, location, starttime, endtime, slots, is_past, eventimage) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";
  184     if (err) {
  185       console.error('Error adding event to the database:', err);
  186       return res.status(500).send('Server error');
  187     }
  188     console.log('Event added to the database');
  189     res.status(200).send('Event added successfully');
  190   });
  191 });
  192
  193 app.use('/uploads', express.static('uploads'));
  194
  195 // Assuming you have a database connection established
  196 // Assuming you have a database connection established
  197 // Assuming you have a database connection established
  198 app.post('/registerevent', (req, res) => { "registerevent": Unknown word.
  199   const { name, rollno, email, sec, department, year } = req.body; "rollno": Unknown word.
  200
  201   const query = "INSERT INTO registrations (name, rollno, email, sec, department, year) VALUES (?, ?, ?, ?, ?, ?); " "rollno": Unknown word.
  202   connection.query(query, [name, rollno, email, sec, department, year], (err, result) => { "rollno": Unknown word.
  203     if (err) {
  204       console.error('Error registering student for the event:', err);
  205       return res.status(500).send('Server error');
  206     }
  207   }
  208
  209   // Create email transporter
  210   const transporter = nodemailer.createTransport({
  211     service: 'gmail',
  212     auth: {
  213       user: 'chimut10200@gmail.com', // Your email
  214       pass: 'Sefis1xjg yoxc nca' // App-specific password "sefis": Unknown word.
  215   });
  216
  217   // Email options
  218   const mailOptions = {
  219     from: 'chimut10200@gmail.com', // Sender address
  220     to: email, // Recipient's email address
  221     subject: 'Registration Successful',
  222     text: `Hello ${name},\n\n${you have successfully registered for the event.\n\nHere are your details:\nName: ${name}\nRoll No: ${rollno}\nSection: ${sec}\nDepartment: ${department}`};
  223
  224   // Send email
  225   transporter.sendMail(mailOptions, (error, info) => {
  226     if (error) {
  227       console.log('Error occurred while sending email:', error);
  228       return res.status(500).send(`Error sending email: ${error.message}`);
  229     } else {
  230       console.log('Email sent: ' + info.response);
  231       res.status(200).send('Registration successful and email sent');
  232     }
  233   });
  234 });
  235
  236 const multer = require('multer');
  237 //const path = require('path');
  238
  239 // Set storage engine
  240 const storage = multer.diskStorage({
  241   destination: './uploads/',
  242   filename: (req, file, cb) => {
  243     cb(null, file.filename + '-' + Date.now() + path.extname(file.originalname)); "filename": Unknown word.
  244   }
  245 });
  246
  247
  248
  249
  250
  251
  252
  253
  254
  255
  256
  257
  258
  259
  260
  261
  262
  263
  264
  265
  266
  267
  268
  269
  270
  271
  272
  273
  274
  275
  276
  277
  278
  279
  280
  281
  282
  283
  284
  285
  286
  287
  288
  289
  290
  291
  292
  293
  294
  295
  296
  297
  298
  299
  300
  301
  302
  303
  304
  305
  306
  307
  308
  309
  310
  311
  312
  313
  314
  315
  316
  317
  318
  319
  320
  321
  322
  323
  324
  325
  326
  327
  328
  329
  330
  331
  332
  333
  334
  335
  336
  337
  338
  339
  340
  341
  342
  343
  344
  345
  346
  347
  348
  349
  350
  351
  352
  353
  354
  355
  356
  357
  358
  359
  360
  361
  362
  363
  364
  365
  366
  367
  368
  369
  370
  371
  372
  373
  374
  375
  376
  377
  378
  379
  380
  381
  382
  383
  384
  385
  386
  387
  388
  389
  390
  391
  392
  393
  394
  395
  396
  397
  398
  399
  400
  401
  402
  403
  404
  405
  406
  407
  408
  409
  410
  411
  412
  413
  414
  415
  416
  417
  418
  419
  420
  421
  422
  423
  424
  425
  426
  427
  428
  429
  430
  431
  432
  433
  434
  435
  436
  437
  438
  439
  440
  441
  442
  443
  444
  445
  446
  447
  448
  449
  450
  451
  452
  453
  454
  455
  456
  457
  458
  459
  460
  461
  462
  463
  464
  465
  466
  467
  468
  469
  470
  471
  472
  473
  474
  475
  476
  477
  478
  479
  480
  481
  482
  483
  484
  485
  486
  487
  488
  489
  490
  491
  492
  493
  494
  495
  496
  497
  498
  499
  500
  501
  502
  503
  504
  505
  506
  507
  508
  509
  510
  511
  512
  513
  514
  515
  516
  517
  518
  519
  520
  521
  522
  523
  524
  525
  526
  527
  528
  529
  530
  531
  532
  533
  534
  535
  536
  537
  538
  539
  540
  541
  542
  543
  544
  545
  546
  547
  548
  549
  550
  551
  552
  553
  554
  555
  556
  557
  558
  559
  560
  561
  562
  563
  564
  565
  566
  567
  568
  569
  570
  571
  572
  573
  574
  575
  576
  577
  578
  579
  580
  581
  582
  583
  584
  585
  586
  587
  588
  589
  590
  591
  592
  593
  594
  595
  596
  597
  598
  599
  600
  601
  602
  603
  604
  605
  606
  607
  608
  609
  610
  611
  612
  613
  614
  615
  616
  617
  618
  619
  620
  621
  622
  623
  624
  625
  626
  627
  628
  629
  630
  631
  632
  633
  634
  635
  636
  637
  638
  639
  640
  641
  642
  643
  644
  645
  646
  647
  648
  649
  650
  651
  652
  653
  654
  655
  656
  657
  658
  659
  660
  661
  662
  663
  664
  665
  666
  667
  668
  669
  670
  671
  672
  673
  674
  675
  676
  677
  678
  679
  680
  681
  682
  683
  684
  685
  686
  687
  688
  689
  690
  691
  692
  693
  694
  695
  696
  697
  698
  699
  700
  701
  702
  703
  704
  705
  706
  707
  708
  709
  710
  711
  712
  713
  714
  715
  716
  717
  718
  719
  720
  721
  722
  723
  724
  725
  726
  727
  728
  729
  730
  731
  732
  733
  734
  735
  736
  737
  738
  739
  740
  741
  742
  743
  744
  745
  746
  747
  748
  749
  750
  751
  752
  753
  754
  755
  756
  757
  758
  759
  760
  761
  762
  763
  764
  765
  766
  767
  768
  769
  770
  771
  772
  773
  774
  775
  776
  777
  778
  779
  780
  781
  782
  783
  784
  785
  786
  787
  788
  789
  790
  791
  792
  793
  794
  795
  796
  797
  798
  799
  800
  801
  802
  803
  804
  805
  806
  807
  808
  809
  810
  811
  812
  813
  814
  815
  816
  817
  818
  819
  820
  821
  822
  823
  824
  825
  826
  827
  828
  829
  830
  831
  832
  833
  834
  835
  836
  837
  838
  839
  840
  841
  842
  843
  844
  845
  846
  847
  848
  849
  850
  851
  852
  853
  854
  855
  856
  857
  858
  859
  860
  861
  862
  863
  864
  865
  866
  867
  868
  869
  870
  871
  872
  873
  874
  875
  876
  877
  878
  879
  880
  881
  882
  883
  884
  885
  886
  887
  888
  889
  890
  891
  892
  893
  894
  895
  896
  897
  898
  899
  900
  901
  902
  903
  904
  905
  906
  907
  908
  909
  910
  911
  912
  913
  914
  915
  916
  917
  918
  919
  920
  921
  922
  923
  924
  925
  926
  927
  928
  929
  930
  931
  932
  933
  934
  935
  936
  937
  938
  939
  940
  941
  942
  943
  944
  945
  946
  947
  948
  949
  950
  951
  952
  953
  954
  955
  956
  957
  958
  959
  960
  961
  962
  963
  964
  965
  966
  967
  968
  969
  970
  971
  972
  973
  974
  975
  976
  977
  978
  979
  980
  981
  982
  983
  984
  985
  986
  987
  988
  989
  990
  991
  992
  993
  994
  995
  996
  997
  998
  999
  1000
  1001
  1002
  1003
  1004
  1005
  1006
  1007
  1008
  1009
  1010
  1011
  1012
  1013
  1014
  1015
  1016
  1017
  1018
  1019
  1020
  1021
  1022
  1023
  1024
  1025
  1026
  1027
  1028
  1029
  1030
  1031
  1032
  1033
  1034
  1035
  1036
  1037
  1038
  1039
  1040
  1041
  1042
  1043
  1044
  1045
  1046
  1047
  1048
  1049
  1050
  1051
  1052
  1053
  1054
  1055
  1056
  1057
  1058
  1059
  1060
  1061
  1062
  1063
  1064
  1065
  1066
  1067
  1068
  1069
  1070
  1071
  1072
  1073
  1074
  1075
  1076
  1077
  1078
  1079
  1080
  1081
  1082
  1083
  1084
  1085
  1086
  1087
  1088
  1089
  1090
  1091
  1092
  1093
  1094
  1095
  1096
  1097
  1098
  1099
  1100
  1101
  1102
  1103
  1104
  1105
  1106
  1107
  1108
  1109
  1110
  1111
  1112
  1113
  1114
  1115
  1116
  1117
  1118
  1119
  1120
  1121
  1122
  1123
  1124
  1125
  1126
  1127
  1128
  1129
  1130
  1131
  1132
  1133
  1134
  1135
  1136
  1137
  1138
  1139
  1140
  1141
  1142
  1143
  1144
  1145
  1146
  1147
  1148
  1149
  1150
  1151
  1152
  1153
  1154
  1155
  1156
  1157
  1158
  1159
  1160
  1161
  1162
  1163
  1164
  1165
  1166
  1167
  1168
  1169
  1170
  1171
  1172
  1173
  1174
  1175
  1176
  1177
  1178
  1179
  1180
  1181
  1182
  1183
  1184
  1185
  1186
  1187
  1188
  1189
  1190
  1191
  1192
  1193
  1194
  1195
  1196
  1197
  1198
  1199
  1200
  1201
  1202
  1203
  1204
  1205
  1206
  1207
  1208
  1209
  1210
  1211
  1212
  1213
  1214
  1215
  1216
  1217
  1218
  1219
  1220
  1221
  1222
  1223
  1224
  1225
  1226
  1227
  1228
  1229
  1230
  1231
  1232
  1233
  1234
  1235
  1236
  1237
  1238
  1239
  1240
  1241
  1242
  1243
  1244
  1245
  1246
  1247
  1248
  1249
  1250
  1251
  1252
  1253
  1254
  1255
  1256
  1257
  1258
  1259
  1260
  1261
  1262
  1263
  1264
  1265
  1266
  1267
  1268
  1269
  1270
  1271
  1272
  1273
  1274
  1275
  1276
  1277
  1278
  1279
  1280
  1281
  1282
  1283
  1284
  1285
  1286
  1287
  1288
  1289
  1290
  1291
  1292
  1293
  1294
  1295
  1296
  1297
  1298
  1299
  1300
  1301
  1302
  1303
  1304
  1305
  1306
  1307
  1308
  1309
  1310
  1311
  1312
  1313
  1314
  1315
  1316
  1317
  1318
  1319
  1320
  1321
  1322
  1323
  1324
  1325
  1326
  1327
  1328
  1329
  1330
  1331
  1332
  1333
  1334
  1335
  1336
  1337
  1338
  1339
  1340
  1341
  1342
  1343
  1344
  1345
  1346
  1347
  1348
  1349
  1350
  1351
  1352
  1353
  1354
  1355
  1356
  1357
  1358
  1359
  1360
  1361
  1362
  1363
  1364
  1365
  1366
  1367
  1368
  1369
  1370
  1371
  1372
  1373
  1374
  1375
  1376
  1377
  1378
  1379
  1380
  1381
  1382
  1383
  1384
  1385
  1386
  1387
  1388
  1389
  1390
  1391
  1392
  1393
  1394
  1395
  1396
  1397
  1398
  1399
  1400
  1401
  1402
  1403
  1404
  1405
  1406
  1407
  1408
  1409
  1410
  1411
  1412
  1413
  1414
  1415
  1416
  1417
  1418
  1419
  1420
  1421
  1422
  1423
  1424
  1425
  1426
  1427
  1428
  1429
  1430
  1431
  1432
  1433
  1434
  1435
  1436
  1437
  1438
  1439
  1440
  1441
  1442
  1443
  1444
  1445
  1446
  1447
  1448
  1449
  1450
  1451
  1452
  1453
  1454
  1455
  1456
  1457
  1458
  1459
  1460
  1461
  1462
  1463
  1464
  1465
  1466
  1467
  1468
  1469
  1470
  1471
  1472
  1473
  1474
  1475
  1476
  1477
  1478
  1479
  1480
  1481
  1482
  1483
  1484
  1485
  1486
  1487
  1488
  1489
  1490
  1491
  1492
  1493
  1494
  1495
  1496
  1497
  1498
  1499
  1500
  1501
  1502
  1503
  1504
  1505
  1506
  1507
  1508
  1509
  1510
  1511
  1512
  1513
  1514
  1515
  1516
  1517
  1518
  1519
  1520
  1521
  1522
  1523
  1524
  1525
  1526
  1527
  1528
  1529
  1530
  1531
  1532
  1533
  1534
  1535
  1536
  1537
  1538
  1539
  1540
  1541
  1542
  1543
  1544
  1545
  1546
  1547
  1548
  1549
  1550
  1551
  1552
  1553
  1554
  1555
  1556
  1557
  1558
  1559
  1560
  1561
  1562
  1563
  1564
  1565
  1566
  1567
  1568
  1569
  1570
  1571
  1572
  1573
  1574
  1575
  1576
  1577
  1578
  1579
  1580
  1581
  1582
  1583
  1584
  1585
  1586
  1587
  1588
  1589
  1590
  1591
  1592
  1593
  1594
  1595
  1596
  1597
  1598
  1599
  1600
  1601
  1602
  1603
  1604
  1605
  1606
  1607
  1608
  1609
  1610
  1611
  1612
  1613
  1614
  1615
  1616
  1617
  1618
  1619
  1620
  1621
  1622
  1623
  1624
  1625
  1626
  1627
  1628
  1629
  1630
  1631
  1632
  1633
  1634
  1635
  1636
  1637
  1638
  1639
  1640
  1641
  1642
  1643
  1644
  1645
  1646
  1647
  1648
  1649
  1650
  1651
  1652
  1653
  1654
  1655
  1656
  1657
  1658
  1659
  1660
  1661
  1662
  1663
  1664
  1665
  1666
  1667
  1668
  1669
  1670
  1671
  1672
  1673
  1674
  1675
  1676
  1677
  1678
  1679
  1680
  1681
  1682
  1683
  1684
  1685
  1686
  1687
  1688
  1689
  1690
  1691
  1692
  1693
  1694
  1695
  1696
  1697
  1698
  1699
  1700
  1701
  1702
  1703
  1704
  1705
  1706
  1707
  1708
  1709
  1710
  1711
  1712
  1713
  1714
  1715
  1716
  1717
  1718
  1719
  1720
  1721
  1722
  1723
  1724
  1725
  1726
  1727
  1728
  1729
  1730
  1731
  1732
  1733
  1734
  1735
  1736
  1737
  1738
  1739
  1740
  1741
  1742
  1743
  1744
  1745
  1746
  1747
  1748
  1749
  1750
  1751
  1752
  1753
  1754
  1755
  1756
  1757
  1758
  1759
  1760
  1761
  1762
  1763
  1764
  1765
  1766
  1767
  1768
  1769
  1770
  1771
  1772
  1773
  1774
  1775
  1776
  1777
  1778
  1779
  1780
  1781
  1782
  1783
  1784
  1785
  1786
  1787
  1788
  1789
  1790
  1791
  1792
  1793
  1794
  1795
  1796
  1797
  1798
  1799
  1800
  1801
  1802
  1803
  1804
  1805
  1806
  1807
  1808
  1809
  1810
  1811
  1812
  1813
  1814
  1815
  1816
  1817
  1818
  1819
  1820
  1821
  1822
  1823
  1824
  1825
  
```

RESULTS:

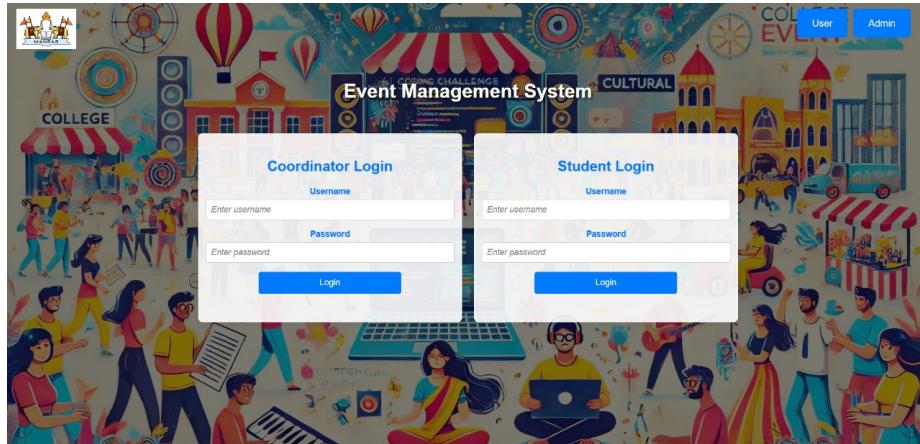


Fig 9: Login page

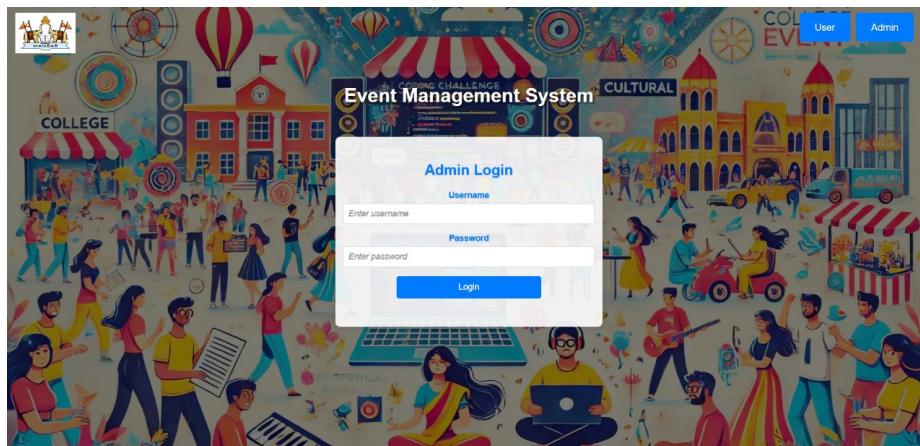


Fig 10: AdminLogin page

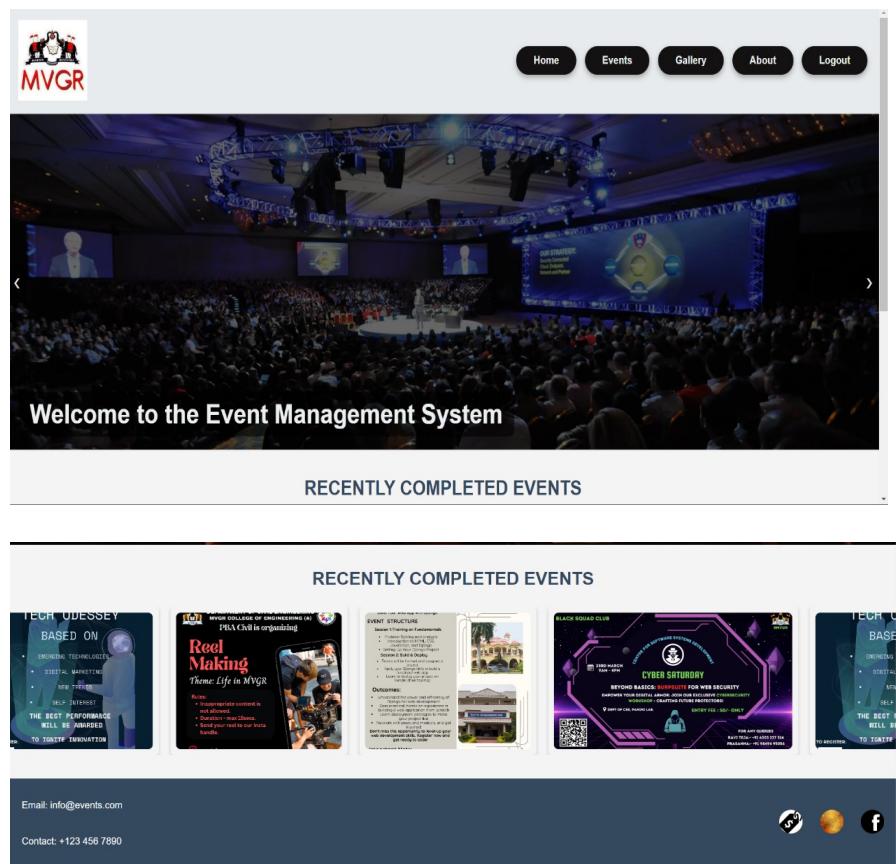


Fig 11:Home page

The screenshot displays a web page titled "Events" with a dark blue header featuring a circuit board pattern. Below the header, there are two main sections: "Upcoming Events" and "Past Events".

Upcoming Events:

- Technical
- Non-Technical
- Cultural
- Sports
- Workshops

Past Events:

- CYBER SATURDAY
- TECH ODYSSEY
- Real Making

Fig 12: Event Page

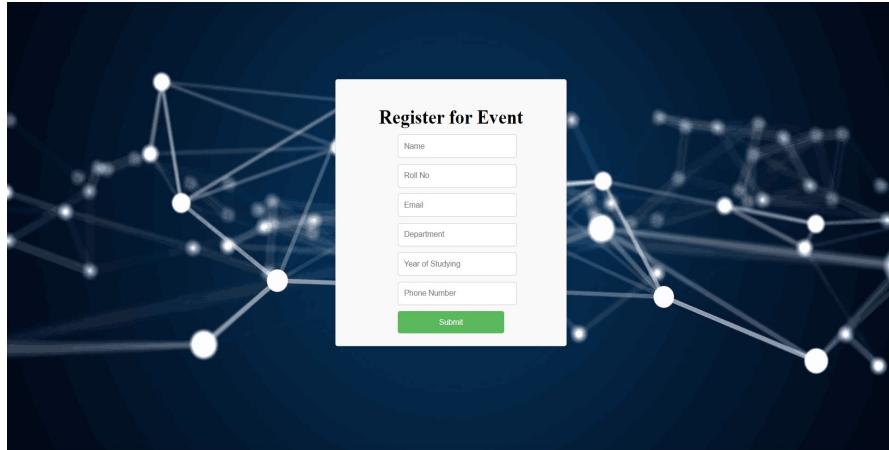


Fig 13:Student event register

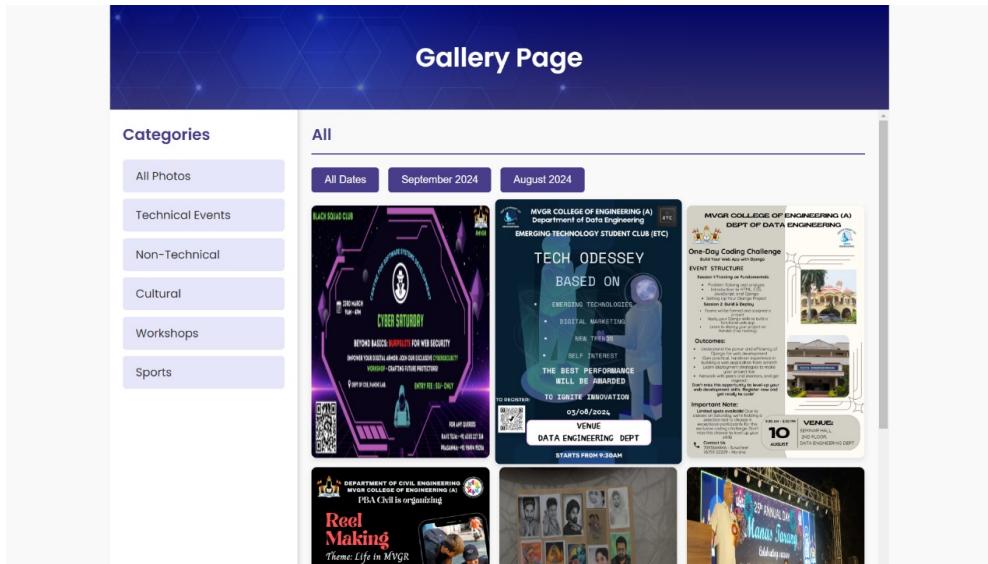


Fig 14:Gallery page

About Us

About Our College Events

Welcome to our college events page! Our college hosts a variety of events throughout the year, including technical and non-technical activities, cultural festivals, workshops, and sports tournaments. These events provide students with opportunities to enhance their skills, engage with peers, and build a sense of community.



Technical Events

Our technical events include coding competitions, hackathons, and seminars on emerging technologies. These events are designed to foster technical skills and innovation among students.

Workshops

We organize workshops on various topics, including skill development, career guidance, and personal growth. These workshops provide practical knowledge and hands-on experience in different areas.



Non-Technical Events

Our non-technical events include quizzes, debates, and literary activities. These events provide a platform for students to showcase their knowledge and communication skills in various non-technical fields.



Sports Tournaments

Our sports tournaments feature a range of athletic competitions, from team sports to individual events. These tournaments encourage students to stay active, work as a team, and demonstrate their athletic abilities.



Cultural Festivals

Our cultural festivals celebrate diversity through music, dance, food, and art. These events are a highlight of the academic year and foster a sense of community and cultural appreciation.



Fig 15 :About us

COORDINATOR SIDE:

The screenshot shows a web-based event management system. At the top, there is a banner with the word "Events" and a background image of a crowded event. Below the banner, there are two sections: "Past Events" and "Upcoming Events". The "Past Events" section contains one item: "Cyber Security" with a brief description. The "Upcoming Events" section has a button labeled "Add Event".

Fig 16: Coordinator-event page

The screenshot shows the "Add New Event" page. It features a large banner image of a speaker on stage in front of a large audience. Below the banner, there are several input fields: "Event Name" (text input), "Description" (text area), "Event Image" (file upload input), "Category" (dropdown menu with "Technical" selected), and "Event Date" (date input). The page also includes fields for "Location", "Start Time", "End Time", and "Number of Slots Available", each with their respective input types. At the bottom right is a blue "Add Event" button.

Fig 17: Add event details

Admin side:

The screenshot shows the Admin Dashboard interface. On the left, there is a sidebar with a dark background and white text. It has a title 'Admin Dashboard' at the top, followed by two buttons: 'Add Coordinator' and 'List Coordinators'. The main area is titled 'Add New Coordinator' and contains four input fields: 'Username', 'Password', 'Email', and 'Phone'. Below these fields is a green button labeled 'Add Coordinator'.

Fig 18:Adminpage

Conclusion and Future Scope:

The College Event Management System successfully streamlines the process of organizing, managing, and participating in college events. By offering a user-friendly interface, seamless event registration, and a visually appealing gallery for past events, the system enhances the overall event experience for both students and administrators. The integration of modern web technologies like Node.js, MySQL, and responsive design ensures efficient performance and accessibility across devices. Moreover, the administrative interface simplifies the task of managing multiple events, reducing the need for manual oversight and minimizing errors in event handling. Overall, this system delivers a robust and scalable solution for college event management, addressing the challenges faced by traditional manual processes.

While the College Event Management System provides a solid foundation, there are several opportunities to enhance its functionality and usability in the future:

Integrating real-time notifications via email or SMS for event reminders, updates, and changes can enhance user engagement by keeping participants informed. Additionally, supporting paid events through a secure payment gateway like PayPal or Stripe would facilitate online ticket purchasing and registration fees. Developing a mobile app version for iOS and Android users would make the system more accessible, especially for students on the go. Allowing users to sync events with personal calendars (Google Calendar, iCal) and receive automated reminders would further improve convenience. An advanced analytics dashboard for administrators to track metrics like registration rates, attendee demographics, and event popularity would aid in decision-making for future events. Finally, implementing an event feedback system where participants can rate and review events will enable organizers to improve event quality based on participant input.

By implementing these future enhancements, the College Event Management System can evolve into a comprehensive platform that not only simplifies event management but also enriches the experience for users and administrators alike.

REFERENCES:

- Journals-Conferences/Blogs/Articles/web-sites:
 - ✓ <https://ieeexplore.ieee.org/document/9985774>
 - ✓ https://www.irjmets.com/uploadedfiles/paper//issue_11_november_2023/45827/final_file_irjmets1699242577.pdf
 - ✓ <https://www.ijarsct.co.in/Paper17045.pdf>
 - ✓ https://www.researchgate.net/publication/373065249_College_Event_Management_System
 - ✓ https://ijresm.com/Vol.3_2020/Vol3_Iss6_June20/IJRESM_V3_I6_49.pdf