

A PROJECT REPORT ON
CSPHERE – A COLLEGE CLUB MANAGEMENT SYSTEM
SUBMITTED IN
STATE BOARD OF TECHNICAL EDUCATION AND TRAINING
PARTIAL FULFILLMENT FOR THE AWARD OF THE DEGREE OF DIPLOMA
IN
COMPUTER ENGINEERING
BY
CH. NAGA VAMSI (23608-CM-008)
G. MANIKANTA (23608-CM-069)
B. VINAY KUMAR (23608-CM-004)
SK. MINAM PALLI JANI BASHA (23608-CM-052)
P. BHARATH (23608-CM-046)
UNDER THE ESTEEMED GUIDANCE OF
K. NAGESWARA RAO _{MTech}



KALLAM HARANADHA REDDY INSTITUTE OF TECHNOLOGY
(AUTONOMOUS)
(APPROVED BY AICTE NEW DELHI, AFFILIATED TO JNTUK-KAKINADA)
ACCREDITED BY NAAC WITH 'A' GRADE AND ACCREDITED BY NBA.
NH-16, CHOWDAVARAM, GUNTUR-522019.
ACADEMIC YEAR – 2025-2026

KALLAM HARANADHA REDDY INSTITUTE OF TECHNOLOGY
(APPROVED BY AICTE NEW DELHI, AFFILIATED TO JNTUK-KAKINADA)
ACCREDITED BY NAAC WITH 'A' GRADE AND ACCREDITED BY NBA
NH-16, CHOWDAVARAM, GUNTUR-522019.

COMPUTER ENGINEERING

CERTIFICATE



This is to certify that the project report entitled “*CSPHERE - A COLLEGE CLUB MANAGEMENT SYSTEM*” is submitted by, **CH. NAGA VAMSI(23608-CM-008), G. MANIKANTA(23608-CM-069), B. VINAY KUMAR(23608-CM-004), P. BHARATH (23608-CM-046), SK. MINAM PALLI JANI BASHA(23608-CM-052)**, to the state board of technical education, Vijayawada in partial fulfilment for the award of degree of Diploma in Computer Engineering is benefited record of project work carried out by them under my supervision during the academic year 2025-2026.

PROJECT GUIDE

K. NAGESWARA RAO MTech

Assistant Professor

HEAD OF THE SECTION

K. NAGESWARA RAO MTech

Assistant Professor

EXTERNAL EXAMINER

DECLARATION

We hereby to declare that the entire project work embodied in this dissertation entitled “**CSPHERE - A COLLEGE CLUB MANAGEMENT SYSTEM**” been independently carried out by us. As per our knowledge, no part of this work has submitted for any degree in any institution, University & organization previously. We hereby boldly state that to the best of my knowledge my work is free from plagiarism.

BATCH NO: 1

- | | |
|-------------------------------|----------------|
| 1. CH. NAGA VAMSI | (23608-CM-008) |
| 2. G. MANIKANTA | (23608-CM-069) |
| 3. B. VINAY KUMAR | (23608-CM-004) |
| 4. SK. MINAM PALLI JANI BASHA | (23608-CM-052) |
| 5. P. BHARATH | (23608-CM-046) |

ACKNOWLEDGEMENT

We proudly express our gratitude and respect towards our honourable chairman is **KALLAM MOHAN REDDY**, chairman of Kallam group for his precious support in the college.

We are thankful to **Dr. M. UMASHANKAR REDDY**, Director, for his encouragement and support for the completion of the project. We are inspired a lot through his valuable message.

We express our immense pleasure to **Dr. B.S.B. REDDY**, Principal, KHIT, GUNTUR, for his support during and till the completion of the project.

We are thankful to Mr. K. NAGESWARA RAO, Head of the Section of CME, for fully providing the laboratory facilities as and when required and for giving us this opportunity to carried out this project work in the college.

We are thankful to our project guide “Mr. K. NAGESWARA RAO”, KHIT, Guntur, for her excellent guidance right from selection of the project and her valuable suggestions throughout this project work.

We are thankful to all teaching and non-teaching staff of the COMPUTER ENGINEERING department and management and my friends for their direct and indirect work provided to us in completing the project.

CH. NAGA VAMSI	(23608-CM-008)
G. MANIKANTA	(23608-CM-069)
B. VINAY KUMAR	(23608-CM-004)
SK. MINAM PALLI JANI BASHA	(23608-CM-052)
P. BHARATH	(23608-CM-046)

ABSTRACT

This project introduces *CSPHERE*, a College Club Management System designed to streamline the organization and participation of student clubs within a campus. The system provides a unified platform where students can explore clubs, join communities, and actively engage through announcements, events, and discussions. It enables club administrators to efficiently manage members, share updates, and coordinate activities, reducing dependency on manual communication and scattered tools. By fostering better interaction and collaboration between clubs and students, CSphere enhances transparency, accessibility, and engagement in extracurricular activities. The platform ultimately contributes to building a more connected and vibrant campus community.

INDEX

CHAPTER NO	NAME OF THE CONTENT	PAGE NO
1	INTRODUCTION	01
2	LITERATURE SURVEY	02
	2.1 Existing System	
	2.2 Proposed System	
3	PROPOSED WORK	03
	3.1 Problem Statement	
	3.2 Objectives	
	3.3 Methodology	
4	SYSTEM REQUIREMENTS SPECIFICATION	05
	4.1 Hardware Requirements	
	4.2 Software Requirements	
	4.3 Modules	
5	SYSTEM DESIGN	07
6	IMPLEMENTATION	10
7	SCREENSHOTS	18
8	CONCLUSION	24
9	FUTURE SCOPE	25
10	REFERENCES	26

LIST OF FIGURES

SERIAL NO	NAME OF THE FIGURE	PAGE NO
1	USECASE DIAGRAM	7
2	CLASS DIAGRAM	8
3	SEQUENCE DIAGRAM	9
4	LANDING PAGE	18
5	SIGNUP	18
6	LOGIN PAGE	19
7	HOME PAGE	19
8	USER PROFILE PAGE	20
9	CHAT PAGE	20
10	CLUB PROFILE PAGE	21
11	EXPLORE PAGE	21
12	ADMIN DASHBOARD PAGE	22
13	ADMIN CLUBS PAGE	22
14	ADMIN EVENTS PAGE	23

CHAPTER 1

INTRODUCTION

INTRODUCTION

Student clubs are an integral part of college life, providing opportunities for students to explore their interests, showcase their talents, and engage in collaborative activities beyond academics. However, the management of these clubs often relies on traditional methods such as manual registrations, printed notices, or fragmented communication through multiple platforms. These methods not only create inefficiencies but also limit student participation and awareness of ongoing activities.

To overcome these challenges, the *College Club Management System (CSphere)* has been developed as a unified digital solution. The system serves as a centralized hub where clubs can publish their activities, events, and announcements, while students can easily discover, join, and participate in the communities of their choice.

The platform provides essential modules such as User Registration and Login, enabling secure access for students and administrators. Students can view event updates, notifications, and discussions in real time, while administrators can efficiently manage memberships, circulate information, and coordinate activities. By integrating these features into a single platform, CSphere reduces the dependency on offline communication and ensures that information reaches the right audience promptly.

The overall aim of the project is to enhance transparency, accessibility, and participation in extracurricular activities. With its structured approach, CSphere not only simplifies the management of clubs but also strengthens the bond between students and campus communities, thereby fostering a more connected and vibrant college environment.

CHAPTER 2

LITERATURE SURVEY

LITERATURE SURVEY

2.1 EXISTING SYSTEM

In most colleges, the management of student clubs is handled through traditional methods such as physical notice boards, word-of-mouth communication, or scattered online groups on social media platforms. These methods often lead to delays in communication, lack of proper documentation, and limited awareness among students about ongoing or upcoming events. Additionally, there is no centralized platform for managing club memberships, tracking participation, or maintaining records of activities. As a result, students miss opportunities to engage, and administrators face difficulties in efficiently coordinating events and keeping members informed.

2.2 PROPOSED SYSTEM

The proposed *College Club Management System (CSphere)* aims to address these limitations by providing a dynamic and interactive platform for both students and administrators. The system will centralize club-related activities such as event announcements, notifications, member registration, and discussions. It will include dedicated user login and registration modules to ensure secure access, along with tools for real-time communication and updates. The platform will be designed to be user-friendly, accessible across devices, and capable of fostering better student engagement. By integrating all club management functionalities under a single system, CSphere enhances transparency, efficiency, and active participation within the campus community.

CHAPTER 3

PROPOSED WORK

PROPOSED WORK

3.1 Problem Statement

The existing methods of managing college clubs rely heavily on manual communication, notice boards, and fragmented online groups, which often result in inefficiency, delays, and lack of awareness among students. There is no centralized platform that allows students to discover clubs, join activities, or stay updated with events in real time. Club administrators also face challenges in managing memberships, circulating announcements, and keeping records. Therefore, there is a need to develop an integrated, user-friendly, and dynamic system that streamlines communication, event coordination, and student participation.

3.2 Objectives

- To develop a centralized digital platform for managing student clubs and their activities.
- To provide secure user registration and login modules for students and administrators.
- To integrate dynamic features such as event management, notifications, and real-time communication.
- To simplify membership management and provide structured records of club activities.
- To ensure scalability, usability, and accessibility across devices.
- To enhance student engagement and foster a connected campus community.

3.3 Methodology

1. Requirement Analysis

- Identify the limitations of existing systems and the specific needs of students, club coordinators, and administrators.
- Conduct surveys or discussions with potential users to gather requirements.
- Define the functional and non-functional requirements of the system.

2. System Design

- Prepare the overall architecture of the system by dividing it into key modules such as User Authentication, Club Management, Event Handling, Notifications, and Real-Time Communication.
- Design database schemas for storing user, event, and club information securely.

Create UI/UX wireframes to ensure smooth navigation and accessibility.

3. Implementation

- Develop the system module by module, starting with authentication and user management, followed by club dashboards, event creation, and notification services.
- Incorporate real-time features such as live chat and instant updates to improve interactivity.
- Apply coding standards and best practices for maintainability and scalability.

4. Integration

- Combine the developed modules into a unified platform.
- Ensure seamless communication between the client side and server side.
- Test interactions between modules such as event updates reflecting instantly in notifications and chat.

5. Testing

- Perform unit testing for individual modules.
- Conduct integration testing to check overall functionality.
- Carry out usability testing with sample users to ensure the system is intuitive.
- Perform security and performance testing to validate reliability.

6. Deployment

- Host the application on a suitable server environment.
- Ensure cross-platform accessibility through responsive design.
- Provide login credentials and role-based access to students and administrators.

7. Maintenance and Future Enhancements

- Regularly monitor system performance and fix bugs.
- Collect feedback from users to introduce improvements.
- Plan for future enhancements such as analytics dashboards, AI-based event recommendations, or mobile app integration.

CHAPTER 4

SYSTEM REQUIREMENTS SPECIFICATION

SYSTEM REQUIREMENT SPECIFICATION

4.1 Hardware Requirements

- **Operating System:** Windows 10 or later
- **Processor:** Multi-core processor (Intel i5 / AMD Ryzen 5 or above)
- **RAM:** 8 GB (recommended for smooth performance)
- **Storage:** 256 GB SSD (for faster data access)
- **Internet Connection:** Stable connection required for testing features

4.2 Software Requirements

- **Front-End:** React.js, Tailwind CSS
- **Back-End:** Node.js with Express.js
- **Database:** MongoDB Atlas
- **Real-time Communication:** Socket.IO
- **Media Storage & Delivery:** Cloudinary
- **Hosting Platform:** Vercel (frontend), Render (backend APIs)
- **Development Tools:** Visual Studio Code, Postman
- **Version Control:** Git & GitHub

4.3 Modules

- **User Authentication Module**
Handles secure registration and login for students and administrators. Supports role-based access control to differentiate between general users, club coordinators, and administrators.
- **Home Page Module**
Provides an overview of the platform, featuring quick links, highlights of active clubs, upcoming events, and announcements. Ensures easy navigation and serves as the entry point for users.
- **Club Management Module**
Allows club administrators to create, update, and manage club profiles. Includes details such as club description, club admins, member list, and achievements. Students can explore and join clubs of interest.

- **Event Management Module**
Enables clubs to schedule, manage, and publicize events. Students can view event details and receive timely updates or reminders.
- **Notification & Announcement Module**
Provides real-time updates about new events, announcements, or changes. Ensures that students and members stay informed without relying on manual communication.
- **Chat & Discussion Module**
Offers real-time messaging among club members for better coordination and engagement. Supports group discussions and activity-related conversations.
- **Search & Explore Module**
Allows students to browse clubs and events using filters or keywords. Improves accessibility by helping users quickly find relevant information.
- **Admin Dashboard Module**
Provides administrators and club heads with tools to manage events, broadcast announcements, and track engagement.

CHAPTER 5

SYSTEM DESIGN

SYSTEM DESIGN

The system design phase defines the overall architecture and functional behaviour of the College Club Management System (CSphere). It translates the requirements into structured models that describe system components, their relationships, and the interactions between users and the system.

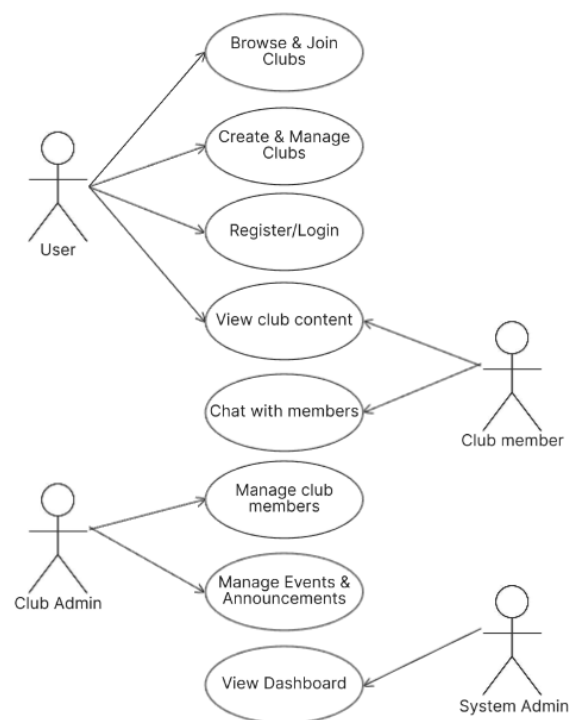
5.1 UML Diagrams

UML diagrams are used to model the system's structure and behaviour. They help in visualizing how different entities (users, admins, clubs, events, etc.) interact and how information flows across the system.

5.1.1 Use Case Diagram

The **Use Case Diagram** illustrates how different actors interact with the system. The key actors include:

- **User:** Can register, login, browse clubs, and join clubs.
- **Club Member:** Can view club content, chat with members, and access announcements and events.
- **Club Admin:** Can manage club members, create/manage events, and post announcements.
- **System Admin:** Oversees the system by viewing statistics and dashboards.

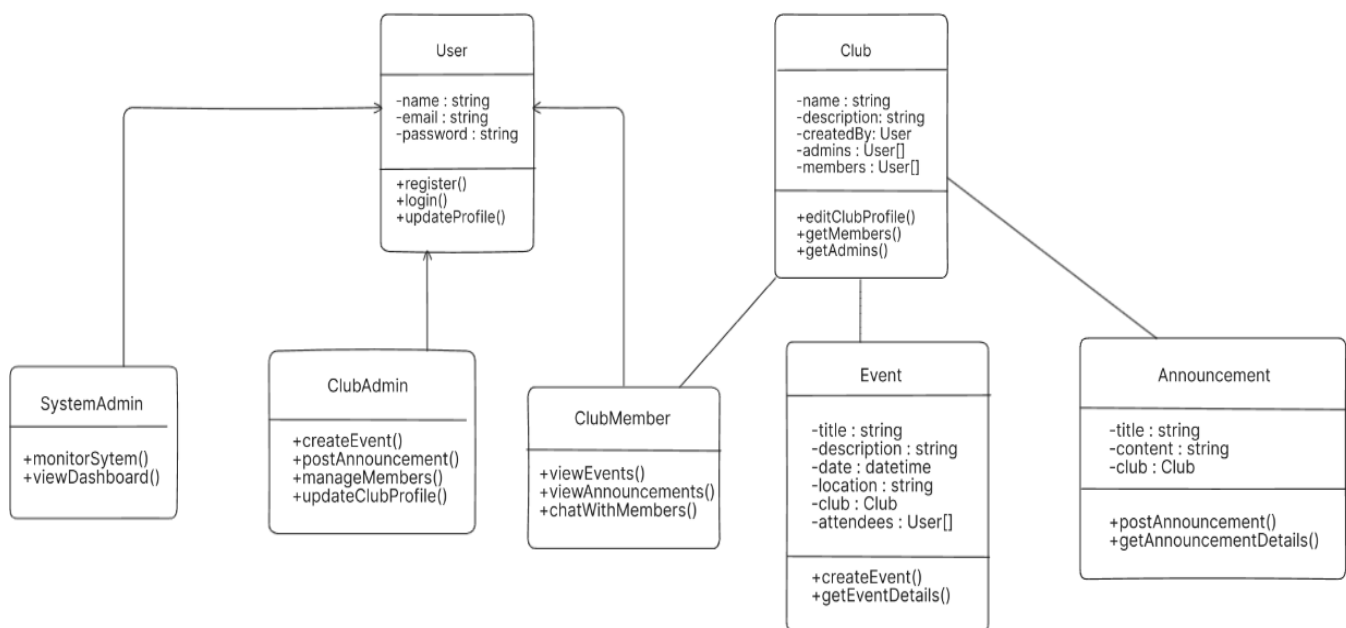


(Ref: Fig 5.1 Use Case Diagram)

5.1.2 Class Diagram

The **Class Diagram** shows the static structure of the system by representing classes, attributes, methods, and relationships between entities.

- **User Class:** Includes attributes like name, email, password, and methods for registration, login, and updating profile.
- **Club Class:** Contains details of clubs (name, description, members, admins) and methods for profile management and member retrieval.
- **Event Class:** Represents club events with attributes like title, date, location, and attendees.
- **Announcement Class:** Stores announcements with attributes like title and content, posted by a club admin.

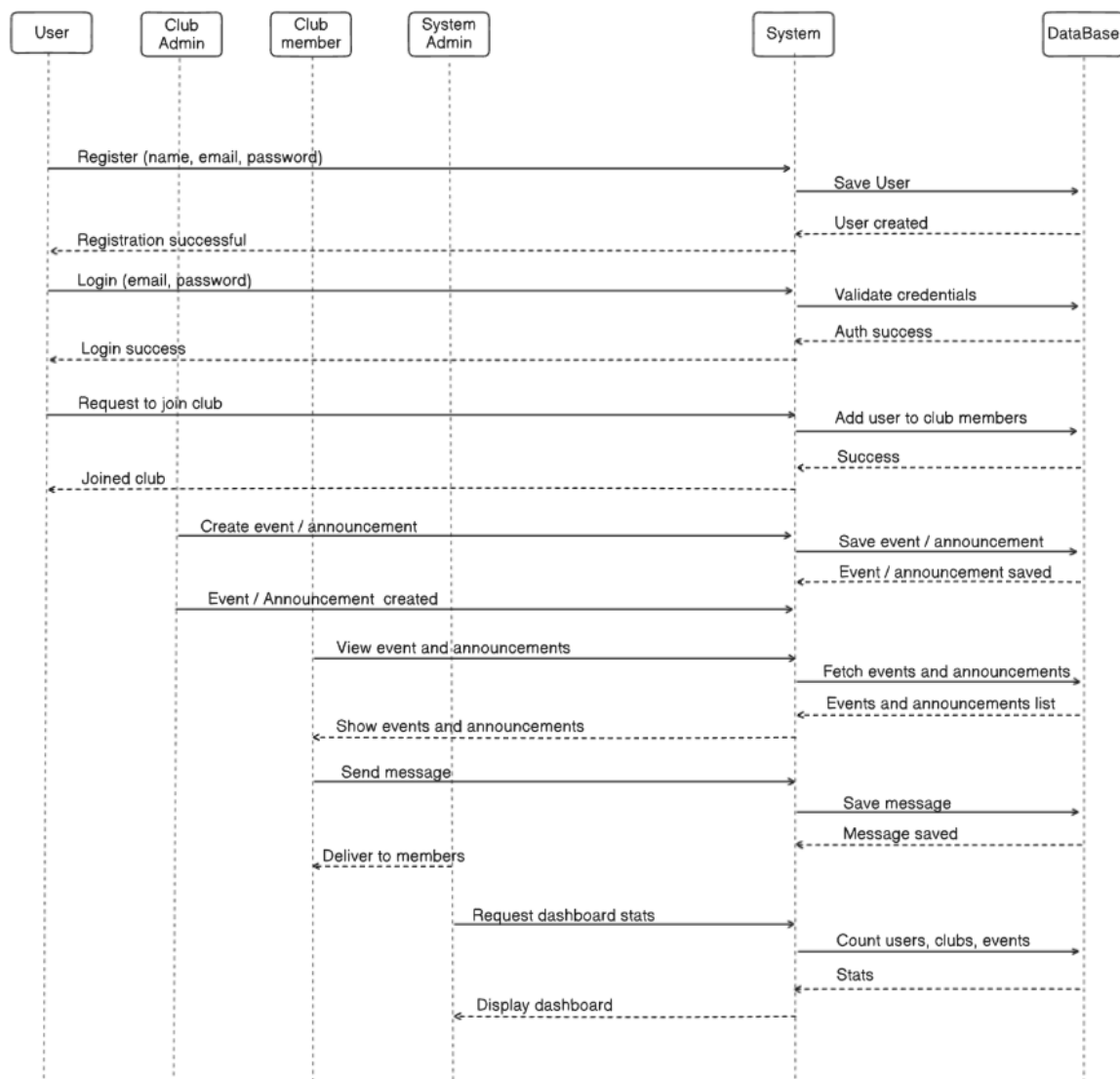


(Ref: Fig 5.2 Class Diagram)

5.1.3 Sequence Diagram

The **Sequence Diagram** captures the dynamic behaviour of the system by showing how operations are executed step by step. It models the flow of actions for:

- **User registration and login** – user submits details → system validates → database saves.
- **Club joining** – user requests to join → system adds to club members list.
- **Event and announcement creation** – club admin submits details → system saves in database → updates shown to members.
- **Viewing events/announcements** – members request → system fetches from database → returns results.
- **System monitoring** – system admin requests dashboard → system queries database → displays statistics.



(Ref: Fig 5.3 Sequence Diagram)

CHAPTER 6

IMPLEMENTATION

IMPLEMENTATION

The implementation phase translates the system design into a functional application. In this project, the *College Club Management System (CSphere)* is developed as a full-stack web application, integrating database models, REST APIs, authentication, real-time communication, and external services such as Cloudinary for media handling. This section highlights key implementation details with representative code snippets.

6.1 User Authentication & Management

The system provides a secure authentication mechanism using JWT (JSON Web Tokens) for session handling and bcrypt for password encryption. Users can register, log in, and update their profile. Additionally, administrators have exclusive privileges to view all registered users.

Code Snippet: User Registration

```
import User from "../models/userModel.js";

import jwt from "jsonwebtoken";

const generateToken = (id) => {

  return jwt.sign({ id }, process.env.JWT_SECRET, { expiresIn: "7d" });

};

export const registerUser = async (req, res) => {

  if (!req.body.name || !req.body.email || !req.body.password) {

    return res.status(400).json({

      success: false,

      message: "All fields are required",

    });

  }

  const name = req.body.name.trim();

  const email = req.body.email.trim();

  const password = req.body.password.trim();

  const profileImageUrl = req.file?.path;
```

```
try {  
  const existingUser = await User.findOne({ email });  
  
  if (existingUser) {  
    return res.status(400).json({  
      success: false,  
      message: "Email is already in use",  
    });  
  }  
  
  const newUser = await User.create({  
    name, email, password, profileImageUrl,  
  });  
  
  res.status(201).json({  
    success: true, message: "User registration successfull",  
    user: newUser, token: generateToken(newUser._id),  
  });  
} catch (error) {  
  console.error("Error in registerUser controller : ", error.message);  
  
  res.status(500).json({  
    success: false,  
    message: "Server Error",  
    error: error.message,  
  });  
}  
};
```


6.2 Club Management

The system enables users to create, join, update, and leave clubs. Each club has members, admins, and a creator, with role-based access control. Only creators can delete clubs, and admins can manage membership and promote/demote members.

Code Snippet: Creating a Club

```
export const createClub = async (req, res) => {  
  const { name, description } = req.body;  
  const userId = req.user._id;  
  
  try {  
    const existingClub = await Club.findOne({ name });  
  
    if (existingClub) {  
      return res.status(400).json({  
        success: false, message: "Club name already taken",  
      });  
    }  
  
    const newClub = await Club.create({  
      name, description, coverImage: req.file?.path,  
      createdBy: userId,  
      members: [userId],  
      admins: [userId],  
    });  
  
    res.status(201).json({  
      success: true, message: "Club creation successfull", club: newClub,  
    });  
  } catch (error) {  
    console.log("Error in createClub controller : ", error.message);  
  
    res.status(500).json({  
      success: false, message: "Error Creating Club", error: error.message,  
    });  
  }  
};
```

Joining a Club

```
export const joinClub = async (req, res) => {  
  const userId = req.user._id;  
  
  try {  
    const club = await Club.findById(req.params.id);  
  
    if (!club) {  
      return res.status(404).json({  
        success: false, message: "Club not found",  
      });  
    }  
  
    if (club.members.includes(userId)) {  
      return res.status(400).json({  
        success: false,  
        message: "Already a member",  
      });  
    }  
  
    club.members.push(userId);  
    await club.save();  
  
    res.status(200).json({  
      success: true, message: "Joined club successfully", club,  
    });  
  } catch (error) {  
    console.log("Error in joinClub controller : ", error.message);  
  
    res.status(500).json({  
      success: false,  
      message: "Error joining club",  
      error: error.message,  
    });  
  }  
};
```

Toggle Admin Role

```
export const toggleAdmin = async (req, res) => {  
  const userId = req.user._id;  
  const { userToToggle } = req.body;  
  try {  
    const club = await Club.findById(req.params.id);  
    if (!club) {  
      return res.status(404).json({  
        success: false, message: "Club not found",  
      });  
    }  
    const isClubAdmin = club.admins.includes(userId.toString());  
    if (!isClubAdmin) {  
      return res.status(403).json({  
        success: false, message: "Not authorized to perform this action",  
      });  
    }  
    const isUserMember = club.members.includes(userToToggle);  
    if (!isUserMember) {  
      return res.status(400).json({  
        success: false, message: "User is not a member of the club",  
      });  
    }  
    const isAlreadyAdmin = club.admins.includes(userToToggle);  
    if (isAlreadyAdmin) {  
      club.admins = club.admins.filter(  
        (adminId) => adminId.toString() !== userToToggle.toString()  
      );  
    } else {  
      club.admins.push(userToToggle);  
    }  
  }  
}
```

```

}

await club.save();

res.status(200).json({

  success: true,

  message: `User ${

    isAlreadyAdmin ? "removed from" : "added as"

  } admin successfully`,

  admins: club.admins,

});

} catch (error) {

  console.error("Error in toggleAdmin controller:", error.message);

  res.status(500).json({

    success: false, message: "Error toggling admin status",

    error: error.message,

  });

}

};

```

Socket Management:

This module manages real-time communication between users through Socket.IO. It keeps track of active users, handles joining and leaving club chat rooms, and enables instant messaging. It also provides custom methods to send notifications and update unread message counts directly to connected users, ensuring real-time interactivity across the platform.

```

const activeUsers = new Map();

export default function socketHandler(io) {

  io.on("connection", (socket) => {

    socket.on("register", (userId) => {

      activeUsers.set(userId, socket.id);

    });

    socket.on("joinRoom", (clubId) => {

      socket.join(clubId)

```

```

});

socket.on("leaveRoom", (clubId) => {
  socket.leave(clubId);
});

socket.on("sendMessage", ({ room, message, sender }) => {
  if (!room || !message || !sender) return;
  socket.to(room).emit("message", {
    message, sender,
    createdAt: new Date().toISOString(),
  });
});

socket.on("disconnect", () => {
  for (const [userId, socketId] of activeUsers.entries()) {
    if (socketId === socket.id) {
      activeUsers.delete(userId);
      console.log(`User ${userId} disconnected`);
      break;
    }
  }
});

io.sendNotification = (userId, data) => {
  const socketId = activeUsers.get(userId);
  if (socketId) {
    io.to(socketId).emit("notification", data);
  }
};

io.updateUnreadCount = (userId) => {
  const socketId = activeUsers.get(userId);
  if (socketId) {

```

```
io.to(socketId).emit("updateUnreadCount");  
  
    }  
  
};  
  
}
```

6.X Full Source Code Availability

For brevity, only key implementation snippets are included in this report. The **complete project source code** with all modules, controllers, routes, and configurations is available on GitHub for reference and review. This ensures transparency, maintainability, and easy collaboration.

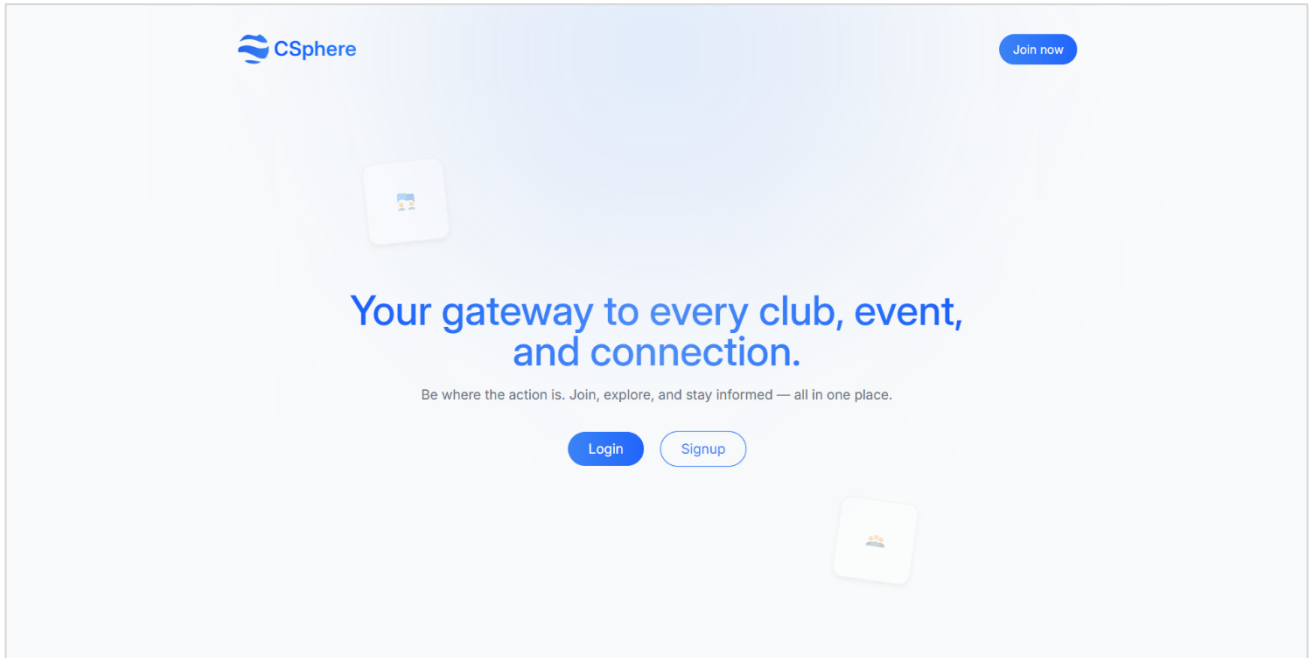
GitHub repository URL : <https://github.com/vamsi-nv/college-project>

CHAPTER 7

SCREENSHOTS

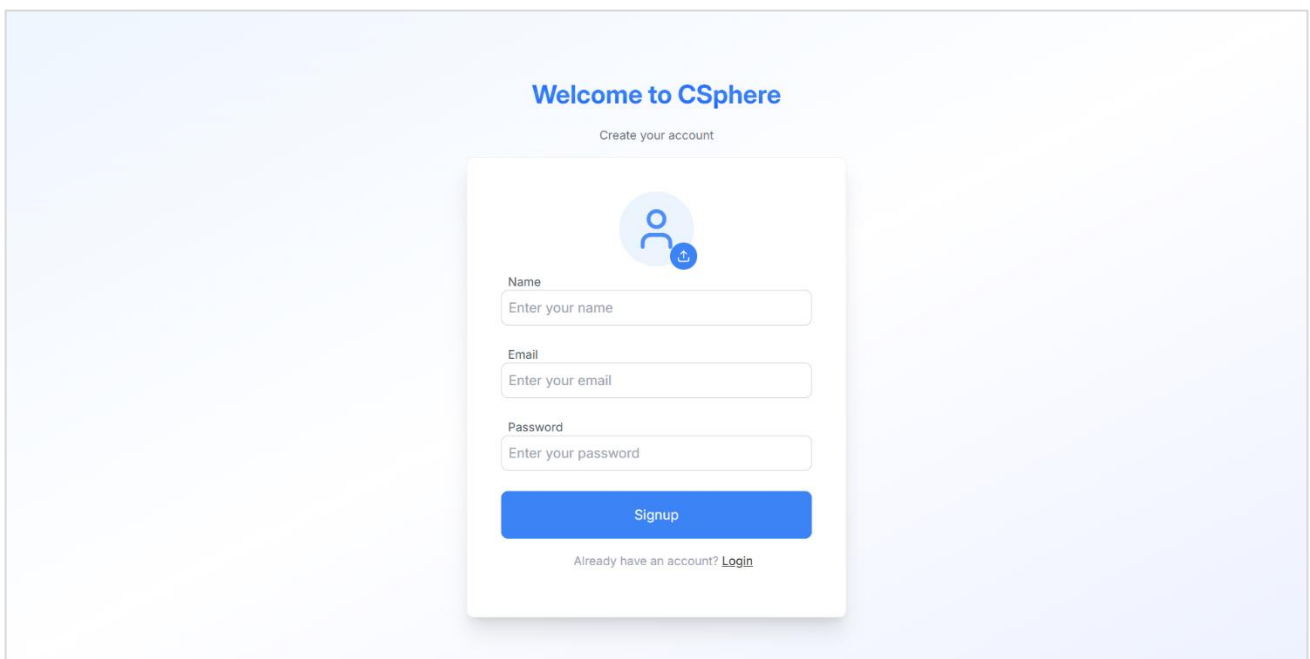
SCREENSHOTS

Landing Page



Ref: Fig 7.1

Signup Page



Ref: Fig 7.2

Login Page

Welcome back

Login to your account

Email Address

Enter your email address

Password

Enter your password

Login

Don't have an account? [Create Account](#)

Ref: Fig 7.3

Home Page

CSphere

Home

Notifications

Explore

Chat

Clubs

Profile

Logout

For You

Events

Announcements

FitLife Sports

Justin • 2 hours ago

Intercollege Football Tournament

Annual football event with teams competing from multiple colleges.

Cancel RSVP

FitLife Sports

Justin • 2 hours ago

Sports Meet Volunteers Needed

Looking for volunteers to help manage the upcoming Intercollege Football Tournament. Sign up at the club office.

CodeCraft

Naga Vamsi • 2 hours ago

Hackathon Registration Open

Registrations for Hack the Future 2025 are now live! Form your teams and register on the club portal.

FitLife Sports

Justin • 2 hours ago

Campus Marathon

A 5K marathon across campus to promote fitness and teamwork.

RSVP

What to join

Euphony Music Club

1 members

Join

New Messages

CodeCraft

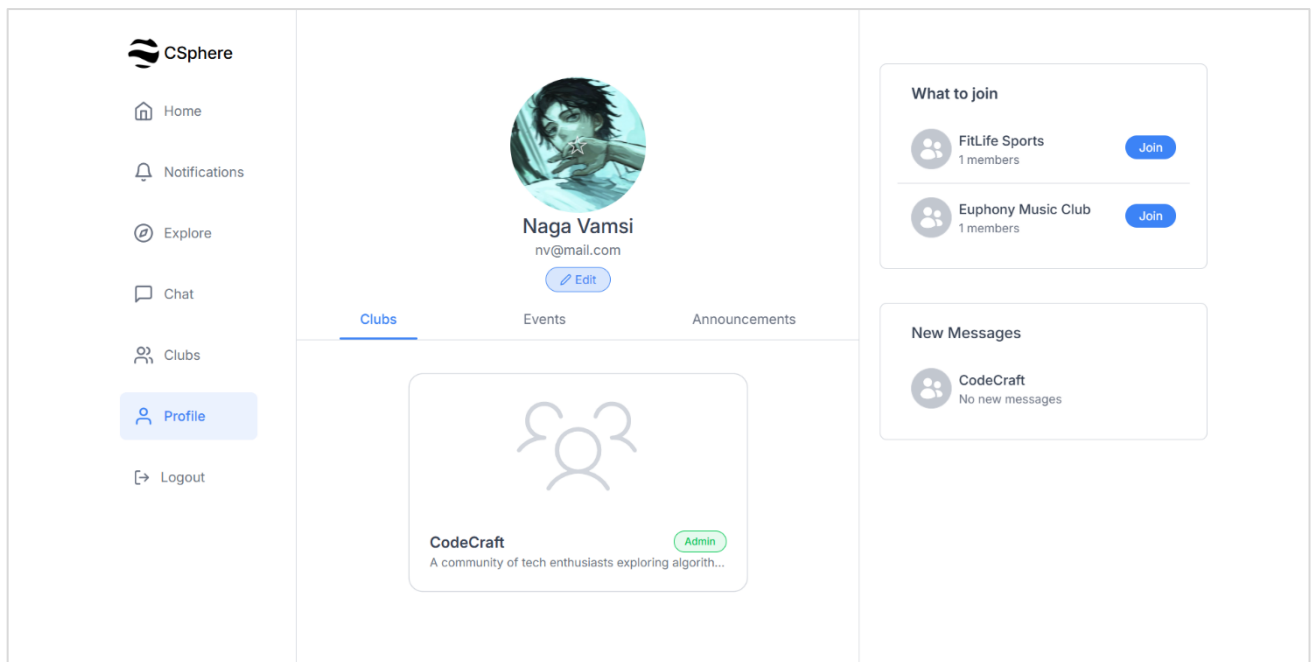
1 new message

FitLife Sports

No new messages

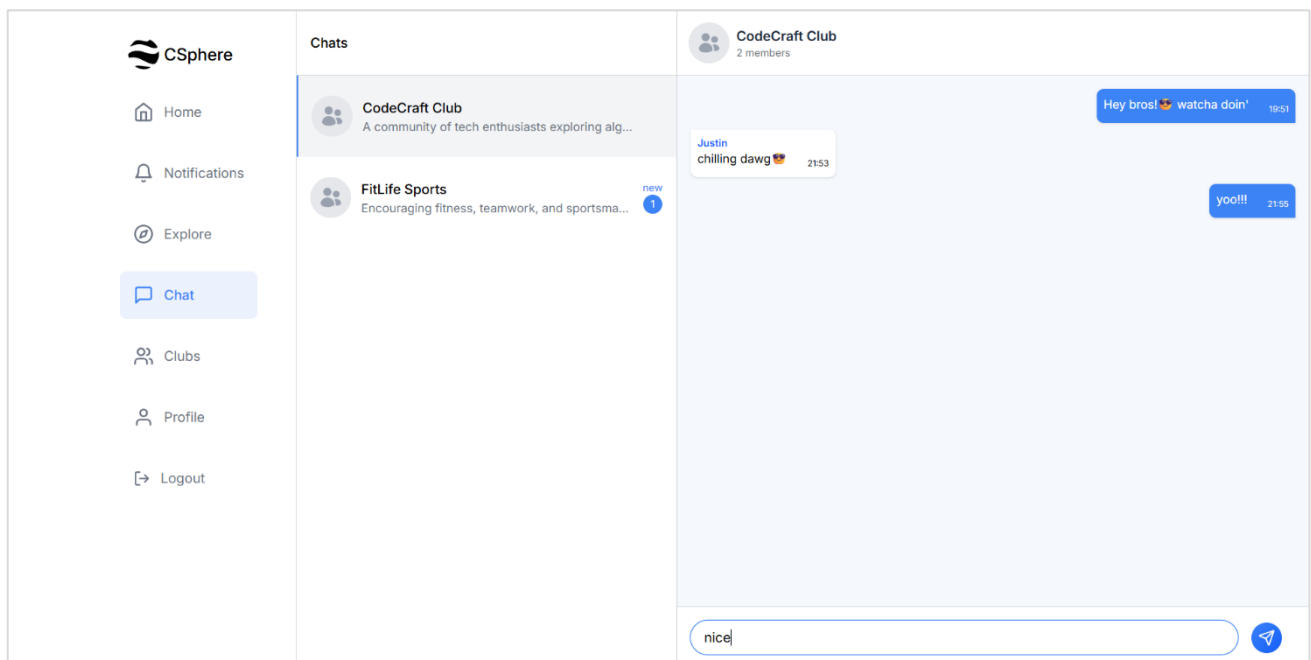
Ref: Fig 7.4

User Profile Page



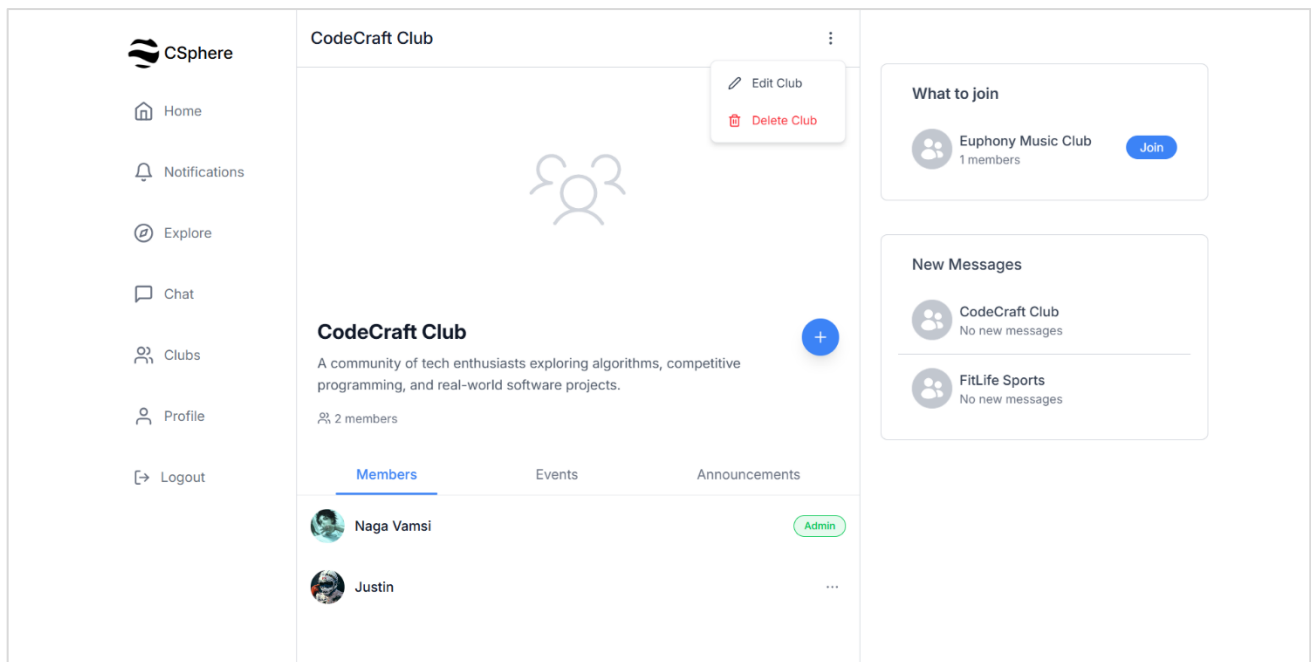
Ref: Fig 7.5

User Chats Page



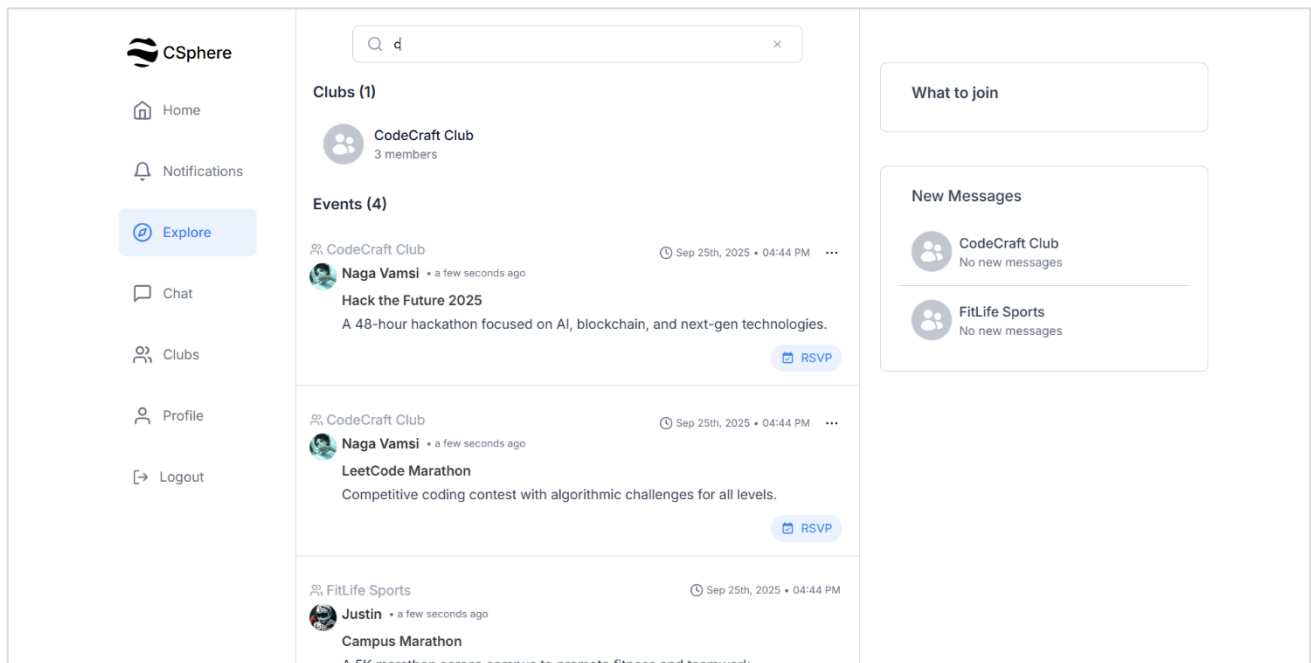
Ref: Fig 7.6

Club Profile Page



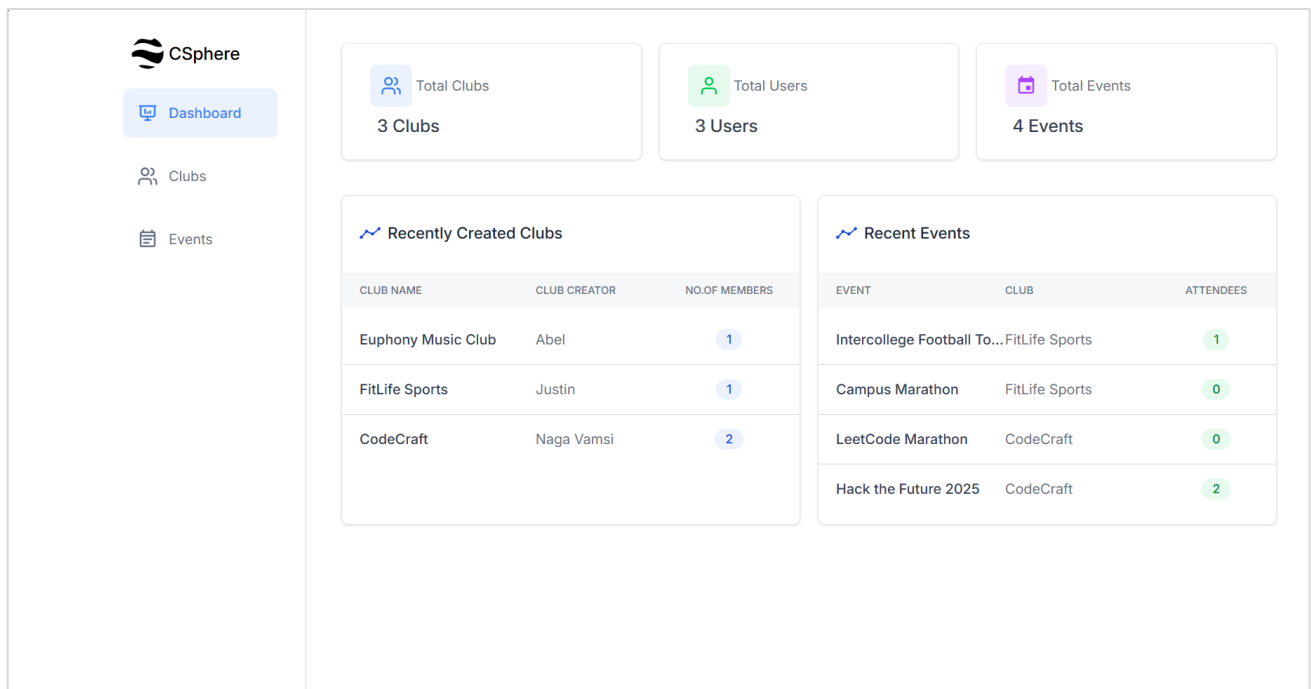
Ref: Fig 7.7

Explore Page



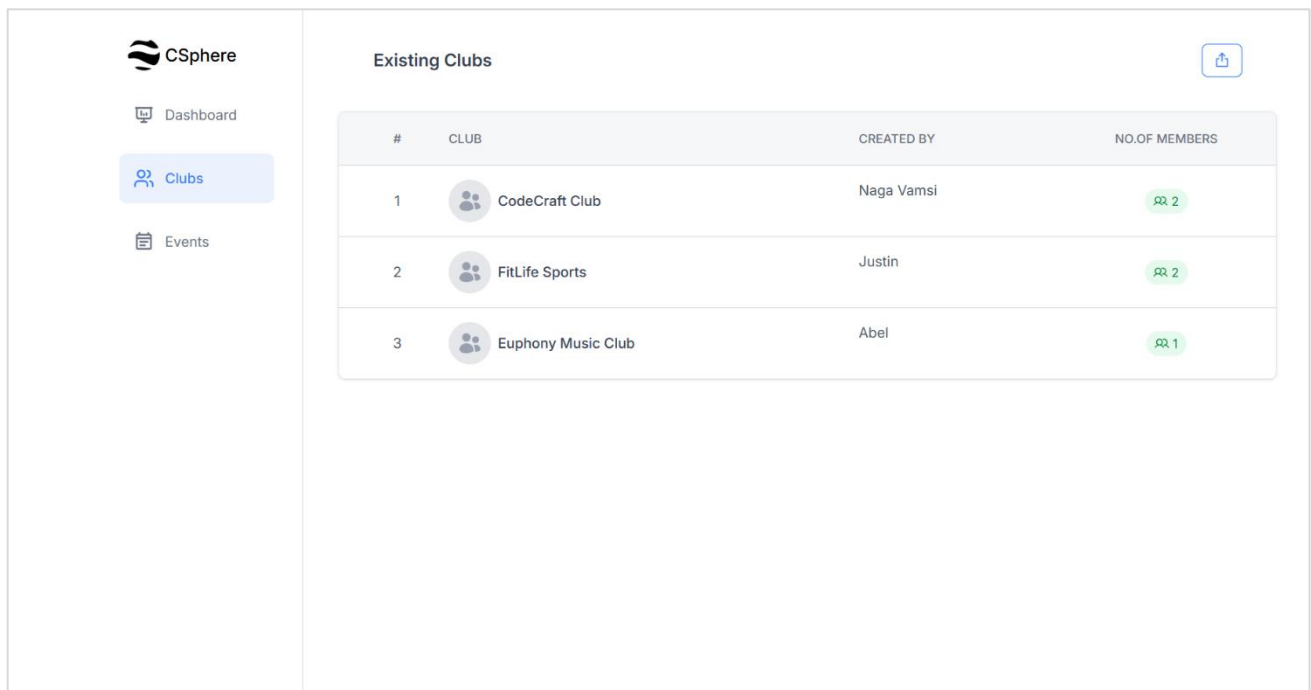
Ref: Fig 7.8

System Administrator Dashboard




Ref: Fig 7.9


System Administrator Clubs list





Ref: Fig 7.10

System Administrator Events list


 CSphere





 Dashboard

 Clubs

 Events

Existing Events



#	EVENT	CLUB	DATE	ATTENDEES
1	Hack the Future 2025	CodeCraft Club	Sep 5, 2025	 2
2	LeetCode Marathon	CodeCraft Club	Sep 15, 2025	 0
3	Campus Marathon	FitLife Sports	Sep 12, 2025	 0
4	Intercollege Football Tournament	FitLife Sports	Sep 25, 2025	 1

Ref: Fig 7.11

CHAPTER 8

CONCLUSION

CONCLUSION

The Club management project (CSphere) has successfully accomplished its goal of building a centralized and user-friendly platform for managing college clubs and fostering student engagement. The system provides streamlined features such as user authentication, club creation and management, event scheduling, announcements, and real-time communication. These functionalities ensure that students, club admins, and system administrators can interact seamlessly within a single digital space.

By integrating role-based access, CSphere ensures that each user type—normal members, club admins, and the system admin—has the appropriate level of control and visibility. The inclusion of real-time chat, notifications, and profile customization enhances collaboration and community building among members.

The system prioritizes security and scalability through JWT-based authentication, database validation, and modular architecture. Additionally, modern deployment considerations make CSphere adaptable to both academic and production environments.

Overall, CSphere not only simplifies the operational aspects of college clubs but also strengthens communication, participation, and transparency across the college community. It represents a step forward in bridging technology with student activities, contributing to a more connected and vibrant campus culture.

CHAPTER 9

FUTURE SCOPE

FUTURE SCOPE

The CSphere project establishes a robust foundation for managing and streamlining college club activities, events, and communication. While the current system provides essential features like club management, event handling, announcements, and real-time chat, there are several enhancements that can be introduced to make the platform even more powerful and engaging. Some possible future scope ideas include:

- **Mobile Application Development:** Launching Android and iOS apps for better accessibility and real-time engagement.
- **Advanced Analytics and Reporting:** Providing insights into student participation, club performance, and event impact.
- **AI-Powered Recommendations:** Suggesting clubs and events to students based on their interests and past activities.
- **Alumni and Community Portal:** Extending access to alumni to build stronger networks and mentorship opportunities.
- **Calendar & Reminder System:** Integrating with personal calendars (Google/Outlook) for event reminders.
- **Gamification Features:** Introducing badges, points, or leaderboards to increase student engagement.
- **Cross-Club Collaborations:** Facilitating joint events and shared resources between multiple clubs.

CHAPTER 10

REFERENCES

REFERENCES

- <https://react.dev>
- <https://nodejs.org>
- <https://expressjs.com>
- <https://socket.io>
- <https://mongoosejs.com>