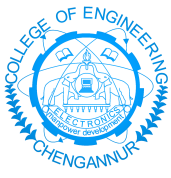
Forums Management System

CSD416 Project Phase II

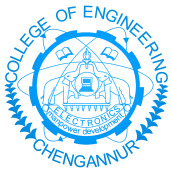
CS20C31 CHN20CS064 Karthik Vijay
CS20C43 CHN20CS085 R Ashwin
CS20C48 CHN20CS097 Sanjay Mathew
B. Tech. Computer Science & Engineering



Department of Computer Engineering
College of Engineering Chengannur
Alappuzha 689121
Phone: +91.479.2455125
http://www.ceconline.edu
hod.cse@ceconline.edu

2024

College Of Engineering Chengannur
Department Of Computer Engineering



CERTIFICATE

This is to certify that, this report titled
Forums Management System
is a bonafide work done by
CS20C31 CHN20CS064 Karthik Vijay
CS20C43 CHN20CS085 R Ashwin
CS20C48 CHN20CS097 Sanjay Mathew

Eighth Semester B.Tech. Computer Science & Engineering students, for the course work in **CSD416 Project Phase II**, under our guidance and supervision, in partial fulfillment of the requirements for the award of the degree, B. Tech. Computer Science & Engineering of **APJ Abdul Kalam Technological University.**

Guide

Guide name
Designation(Assistat or associate Prof...),
Dept. of Computer Engineering
College of Engineering,
Chengannur

Coordinator

Smt. Chinchu M Pillai
Assistant Professor,
Dept. of Computer Engineering
College of Engineering,
Chengannur

External Examiner

Head of the Department

Dr. Manju S Nair
Associate Professor,
Dept. of Computer Engineering
College of Engineering,
Chengannur

Permission to Use

In presenting this project dissertation at College of Engineering Chengannur(CEC) in partial fulfillment of the requirements for an undergraduate degree from APJ Abdul Kalam Technological University, we agree that the libraries of CEC may make it freely available for inspection through any form of media. We further agree that permission for copying of this dissertation in any manner, in whole or in part, for scholarly purposes may be granted by the Head of the Department of Computer Engineering. It is understood that any copying or publication or use of this dissertation or parts thereof for financial gain shall not be allowed without our written permission. It is also understood that due recognition shall be given to us and to CEC in any scholarly use which may be made of any material in this project dissertation.

Karthik Vijay, R Ashwin, Sanjay Mathew

Statement of Authenticity

We hereby declare that this submission is our own work and to the best of our knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at College of Engineering Chengannur(CEC) or any other educational institution, except where due acknowledgement is made in the report. Any contribution made to our work by others, with whom we have worked at CEC or elsewhere, is explicitly acknowledged in the report. We also declare that the intellectual content of this report is the product of our own work done as per the Problem Statement and Proposed Solution sections of the project dissertation report. We have explicitly stated the major references of our work. We have also listed all the documents referred, to the best of our knowledge.

Karthik Vijay, R Ashwin, Sanjay Mathew

Acknowledgement

We take this opportunity to express our deepest sense of gratitude and sincere thanks to everyone who helped us complete this work successfully. We express our sincere thanks to **Dr. Smitha Dharan**, Principal, College Of Engineering, Chengannur and **Dr. Manju S Nair**, Head of Department, Department of Computer Engineering, College of Engineering Chengannur for providing us with all the necessary facilities and support.

We would like to express our sincere gratitude to **Smt. Chinchu M Pillai** , Assistant Professor, Department of Computer Engineering, College of Engineering, Chengannur for their support and cooperation.

We would like to place on record our sincere gratitude to our Project guide, **Sri. Ahammed Siraj K K**, Associate Professor, Department of Computer Engineering, College of Engineering, Chengannur for the guidance and mentorship throughout the course.

Karthik Vijay, R Ashwin, Sanjay Mathew

Abstract

The project proposal encompasses the creation of a comprehensive website designed to effectively manage and monitor various activities within the forums and communities at CEC, including IEEE, IEDC, Tinkerhub, GDSC, TPC, and others. This website not only serves as a centralized hub for these entities but also provides valuable functionality for college staff to oversee forum activities, facilitating accreditation and audit processes. Beyond internal management, the site offers external visibility, enabling visitors to view past and upcoming events. By fostering streamlined management, transparency, and accessibility, the proposed website is poised to significantly enhance the coordination and visibility of college forums, ultimately contributing to a more organized and engaged community.

Contents

Chapter 1

Introduction

This chapter sets the stage for the project, presenting a concise overview of its aims, relevance, and boundaries. It serves as an entry point into the Forums Management System, articulating its role in simplifying forum administration and fostering student involvement within campus communities. Moreover, this chapter contextualizes the project's genesis, elucidating the necessity for a unified platform to streamline forum management and enhance student engagement.

1.1 Proposed Project

To address the challenges posed by the current decentralized forum management at Colleges, we aim to develop a comprehensive web application. This application will serve as a centralized solution, ensuring transparency and an enhanced experience for coordinators and members. The primary focus will be on precise database management and seamless event coordination.

1.1.1 Problem Statement

To develop a web application that centralizes forum management, ensures transparency, and enhances the experience for both coordinators and members, prioritizing precise database management and event coordination.

1.1.2 Proposed Solution

- **Forums Management System (FMS):** A web-based application as a centralized hub for CEC's forums and communities.
- **Key Features:**
  - **Centralized Management:** FMS provides a unified platform for efficient and centralized management of various forums within the CEC community. This includes streamlined administration, user roles, and content moderation.
  - **Event Scheduling:** The system incorporates a comprehensive event scheduling feature, allowing organizers to plan and coordinate events seamlessly. Users can access a shared calendar, ensuring everyone stays informed about upcoming activities.

- **Attendance Tracking:** FMS offers an attendance tracking mechanism, enabling organizers to monitor and record participation in various events. This feature enhances the overall organization of events and ensures accurate attendance records.
- **Communication Hub:** Serving as a dynamic communication hub, FMS facilitates seamless and effective communication among community members. It includes features such as discussion forums, messaging, and announcements to keep everyone connected.
- **Transparency and Visibility:** FMS enhances transparency by providing real-time updates and visibility into community activities. Important information, announcements, and updates are easily accessible, fostering a sense of openness and inclusivity.
- **Member Engagement:** FMS prioritizes member engagement through interactive features, including discussion forums and collaborative spaces. This promotes active participation, knowledge sharing, and the development of a vibrant and engaged community.

Chapter 2

Report of Preparatory Work

The chapter outlines the preparatory steps taken before the actual implementation of the project.

2.1 Literature Review

Under Literature Review, we conducted an extensive exploration of various platforms that significantly informed our understanding of event management and community-driven initiatives. Through our investigation of KonfHub, GDSC Dashboard, and YepDesk, we gained valuable insights into effective event organization strategies, community engagement approaches, and practical event management functionalities. These findings served as crucial benchmarks for shaping the design and functionality of our Forum Management System.

2.1.1 Platform Exploration

In our exploration of various platforms, we delved into three key sources that significantly shaped our understanding of event management and community-driven initiatives:

- **KonfHub:** Our investigation into KonfHub provided valuable insights into the intricacies of event organization. We gleaned in-depth knowledge on effective strategies, participant engagement, and overall event management practices, setting a foundation for our project's approach to hosting and organizing forums.[10]
- **GDSC Dashboard:** The exploration of the Google Developer Student Clubs (GDSC) Dashboard unveiled valuable insights into community-driven initiatives. Understanding the collaborative aspects and member engagement strategies employed by GDSC influenced our design decisions, particularly in fostering a sense of community within our Forum Management System.[11]
- **YepDesk:** YepDesk became a pivotal reference for optimizing our system's event features. By studying the functionalities of YepDesk, we gained practical insights into event scheduling, attendance tracking, and other features that contributed to the refinement of our own event management capabilities.[9]

2.2 System Study Report

To better comprehend the technological landscape, we delved into system studies, analyzing existing frameworks and platforms. Our focus was on understanding the intricacies of Next.js for front-end development, Express.js for server-side operations, and Node.js for overall backend functionality. This systematic study informed our decision-making process and influenced the architecture of our Forums Management System.

To gather valuable insights and firsthand information, we actively engaged with the executive committee (execom) members of each forum within our college. These interactions were pivotal in understanding their needs, challenges, and the existing event database management systems employed by the forums. This direct engagement with the stakeholders provided essential inputs for the design and features of our Forums Management System.

2.3 A Mini Project based on Events Listing

As part of the preparatory work, we actively engaged in a hands-on project-making event organized by Tinkerhub, named "Stackup." In this collaborative endeavour, we participated as a team, leveraging our skills and knowledge to create a practical application—an Event Listing site.

Project Details:

- **Event Name:** Stackup
- **Organizer:** Tinkerhub
- **Objective:** Develop an Event Listing application using modern web technologies.

Technologies Utilized:

- **Frontend:** Hosted on Vercel, we utilized Next.js for the frontend, ensuring a seamless and responsive user interface.[2]
- **Backend:** Hosted on an Azure server, we employed Express.js[5] and Node.js[1]for the server-side logic.

Deployment and Showcase:

The successful deployment of our mini-project showcased our proficiency in utilizing cutting-edge technologies in a real-world scenario. The frontend hosted on Vercel ensured high availability and efficient delivery of content to end users. Simultaneously, the backend hosted on an Azure server, facilitated by Nginx [3], demonstrated our ability to create a robust and scalable server-side architecture.

Project Links:

- **GitHub Repository:** [6]
- **Live Website:** [7]

Learning Outcomes:

Participating in the "Stackup" event not only allowed us to apply theoretical knowledge but also provided valuable insights into collaborative project development, effective use of version control, and deployment strategies. The experience strengthened our understanding of the entire development lifecycle, from ideation to deployment.

This mini-project served as a pivotal experience in preparing for our main project, providing hands-on exposure to the technologies that would later form the foundation of our Forum Management System.

Chapter 3

Project Design

This chapter outlines the architectural blueprint for the Forums Management System, emphasizing efficiency and user-friendliness. It covers web application architecture, hardware and software requirements, offering insights for developers and stakeholders. The clear presentation of design principles, with diagrams and detailed descriptions, lays the groundwork for effective implementation in subsequent development stages.

3.1 Web Application Architecture

Web application architecture is a vital framework that outlines how applications, middleware systems, and databases interact to ensure seamless operation of web applications. It serves as a blueprint for understanding the application's workings, guiding the development process, planning for future growth, troubleshooting issues, and documenting the application's structure. This architecture is crucial in a project report as it provides a comprehensive overview of the application's structure and data flow, thereby facilitating efficient development, robust scalability, and effective problem-solving.

3.1.1 Web Application Architecuture: General Design

The general architecture gives an overview of how the web application framework works, as shown in Figure 3.1. The explanation for the components is listed below:

- **Client Side:** These are the people using the website or application. They interact with the frontend via web browsers, which communicates with the backend to fetch and display data.
- **Server Side:** This includes the "Web Server service" (such as Nginx), "Cloud/Local Hosting services" (like Azure, Nginx), and "Backend Frameworks" (the specific technologies used to build the backend).
- **Database:** This is where data is stored and managed.

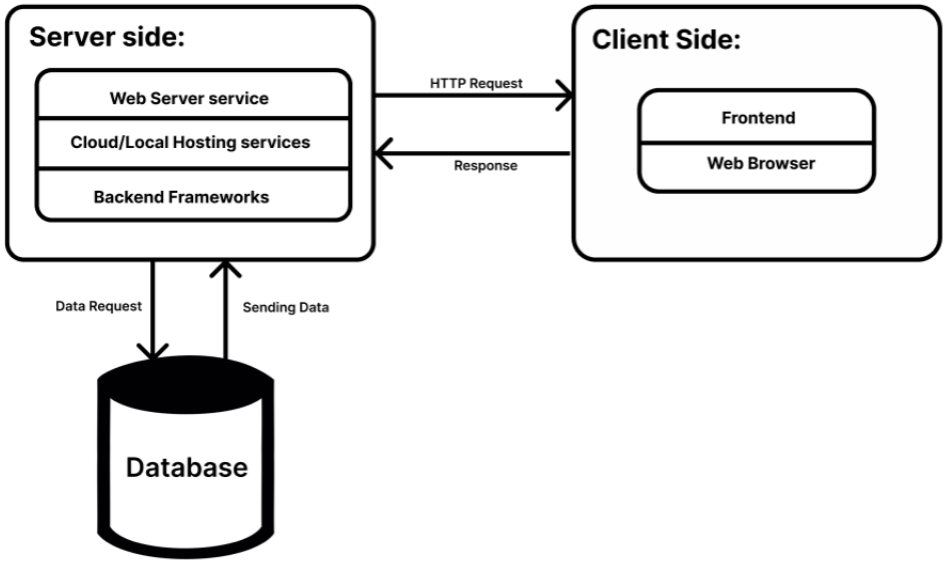


Figure 3.1: Web Application Architecture: General Design

3.1.2 Web Application Architecture: Implementation Design

The subsection outlines the initial design of the web application architecture. It categorizes end users into Institutional Admins, Forum Execom members, and Students. The frontend is built using Next.js for dynamic user interfaces, while the server-side employs Ngrok for secure tunneling, Azure for scalable hosting, and Express.js for efficient application development. MongoDB is chosen as the database for its flexibility and scalability.

End Users:

- Institutional Admins: Responsible for overseeing and managing the overall functioning of the Forum Management System.
- Community Admin: Members of individual forums who act as administrators, organizing events, and managing forum-specific content.
- Students: End users who engage with the system to access information about events, forums, and participate in activities.

Front End: Next.js, a React framework known for its simplicity, efficiency, and seamless client-side rendering, is employed to construct the user interface, ensuring a responsive and dynamic experience. [2]

Server Side: The backend infrastructure utilizes Node.js in conjunction with Express.js, leveraging their robust capabilities for server-side operations. This backend is deployed across two servers, namely server-1 and server-2, ensuring reliability and scalability in handling user requests and data

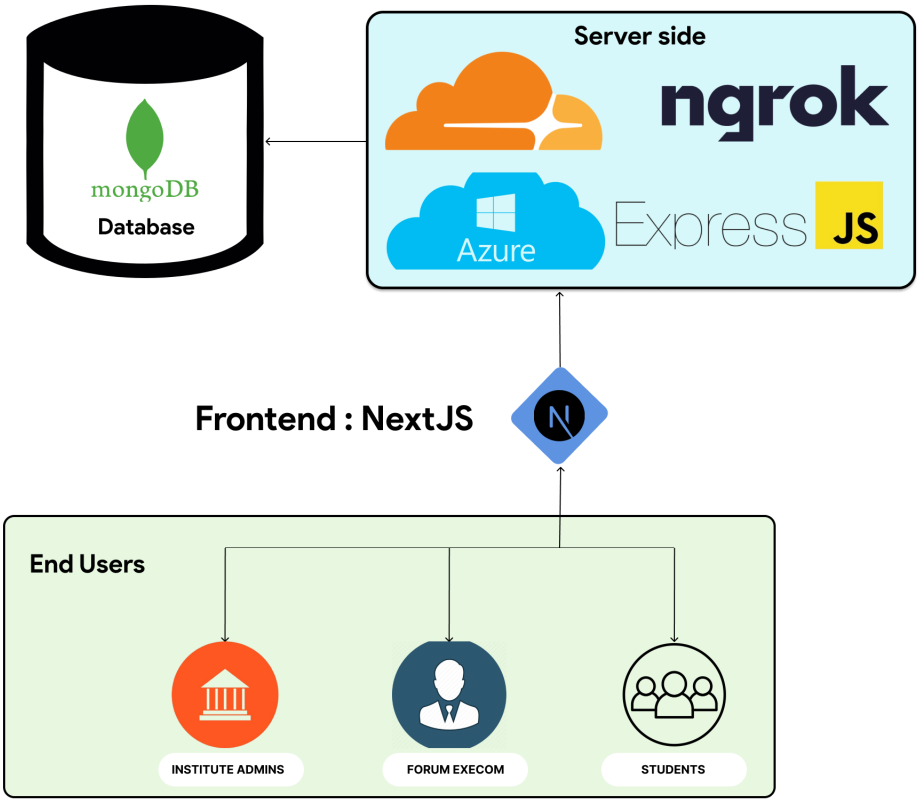


Figure 3.2: Web Application Architecture: Implementation Design

processing. Furthermore, technologies such as Cloudflare and NGINX are integrated into the server configuration, enhancing performance, security, and scalability of the system.

Database: MongoDB, a NoSQL database, is adopted for its flexibility and scalability, making it ideal for managing diverse data structures related to forums, events, and user interactions.

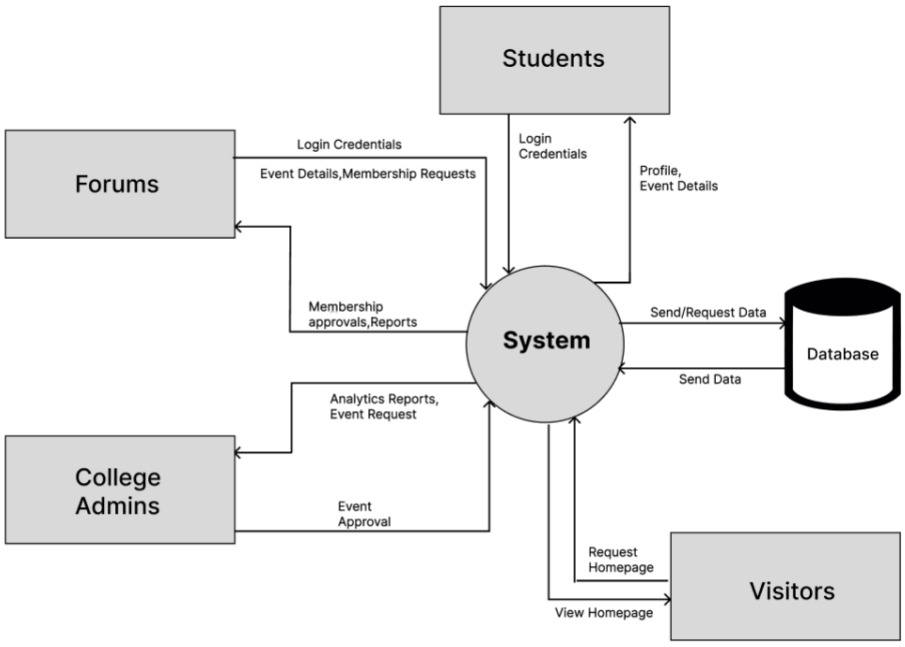


Figure 3.3: Data-Flow Diagram Level-0

3.2 Data-Flow Diagram

One of the key components of our project design is the Data Flow Diagram (DFD). The DFD provides a visual representation of the flow of data within our system. It helps us understand how the system interacts with external entities and how data moves through it.

3.2.1 Level-0

The Level 0 DFD, also known as a context diagram, gives a broad overview of the system. It illustrates how the system interacts with external entities. In our case, these entities are "Students", "Visitors", and "College Admins".

- System: This is the main component of the diagram, representing the system we're designing.
- Data Flows: These are the arrows that show the direction of data flow. They include "Login Credentials", "Event Details", "Membership Reports", "Analytics Reports", "Event Approval", "Send Data", "Request Data", and "View Homepage".
- Database: This represents the system's data storage, where all the necessary data is stored and retrieved from.

This Level 0 DFD is a crucial part of our project design as it provides a high-level understanding of the system's processes and the flow of data. It aids in the clear communication of how the system is designed to work, making it an invaluable tool in our project design documentation.

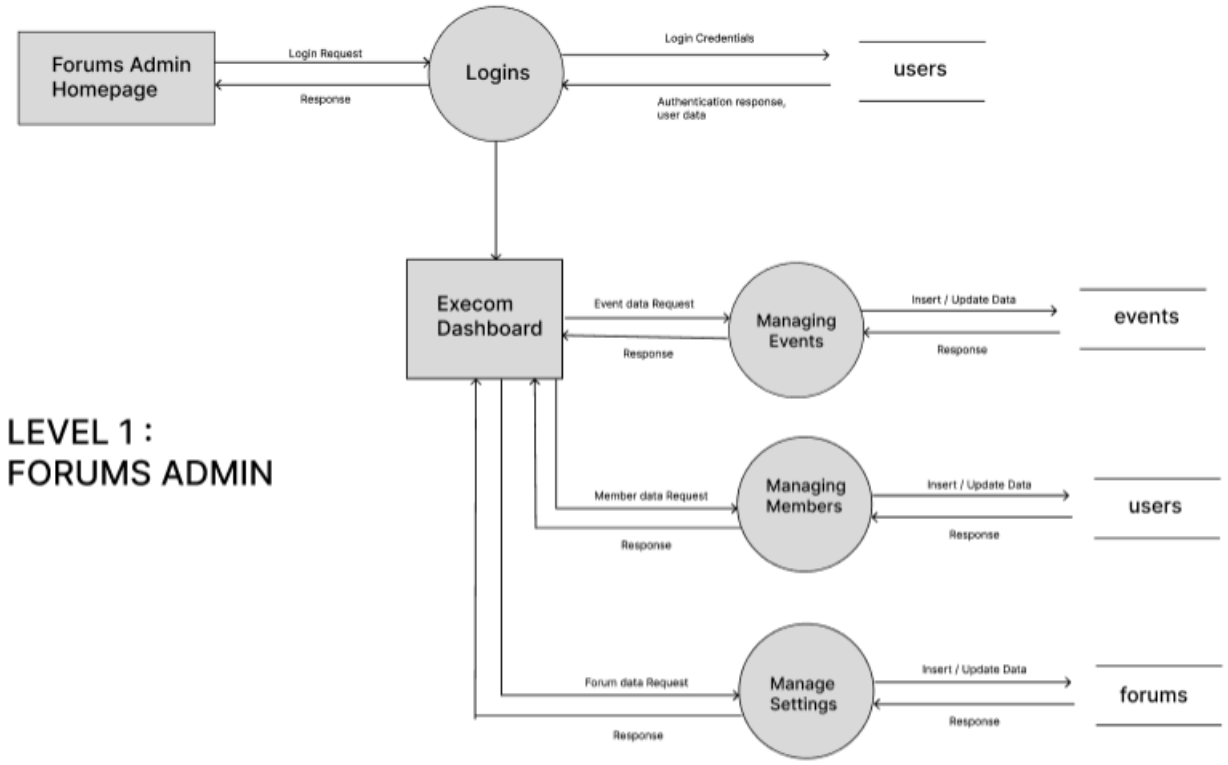


Figure 3.4: Data-Flow Diagram Level-1 Forums

3.2.2 Level-1

A Level 1 Data Flow Diagram (DFD) offers a detailed perspective of a system, breaking down major processes from the Level 0 DFD into sub-processes. Each sub-process is represented as a separate process, displaying associated data flows and data stores. This level provides a nuanced view, illustrating how data flows within the system and interacts with various entities. Essentially, it serves as an "exploded view" of the context diagram, offering insight into the main functions of the system. The choice of DFD level depends on the system's complexity and the desired level of detail, with higher levels providing a broad overview and lower levels delving into specific processes, data flows, and data stores. A combination of different DFD levels ensures a comprehensive understanding of the system.

Forums Admin:

The description of the components in the DFD has listed below:

- **Forums Homepage:** This is where users land when they visit the website where they can view the gallery of the forums in the college.
- **Logins:** This process handles user authentication. When users provide their credentials, this process verifies them and grants access based on their roles.

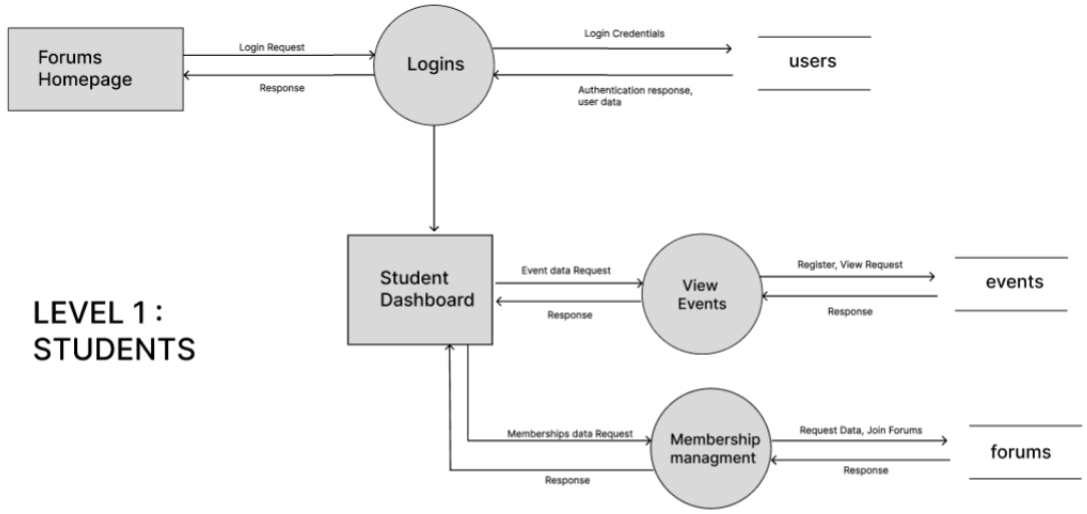


Figure 3.5: Data-Flow Diagram Level-1 Students

- **Users:** This data store contains information about the registered users of the forum.
- **Events:** This data store holds information about the events posted on the forum.
- **Forums:** This data store contains the threads and posts made in the forum.
- **Execom Dashboard:** This is a special dashboard for the executive committee (Execom) or the community admins. It may provide features for managing the forum, such as moderating discussions and managing users.

Students:

- **Forums Homepage:**This is where users land when they visit the website where they can view the gallery of the forums in the college.
- **Student Dashboard:** This is a special dashboard for students. It may provide features for managing the forum, such as moderating discussions and managing users.
- **Logins:** This process handles user authentication. When users provide their credentials, this process verifies them and grants access based on their roles.
- **View Events:** This process involves a user requesting to view event data.
- **Membership Management:** This process involves managing the members of the forum, such as adding new members, updating member information, or removing members.

Institution:

- **Forums Homepage:**This is where users land when they visit the website where they can view the gallery of the forums in the college.

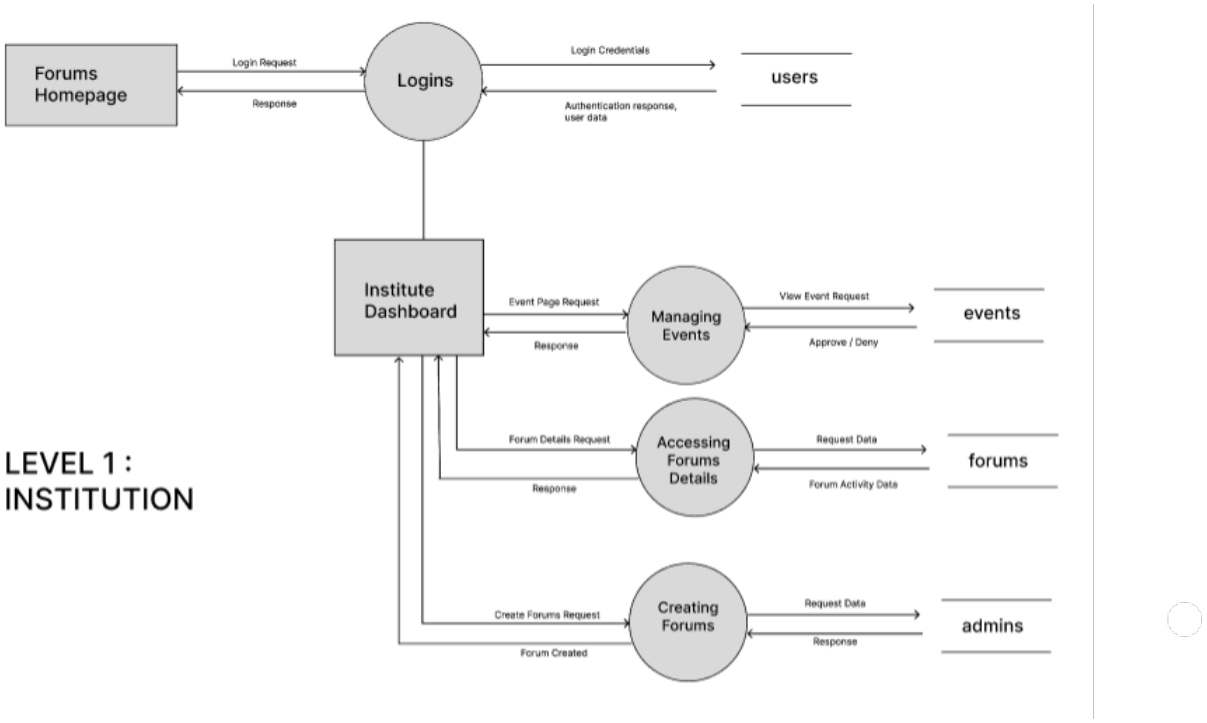


Figure 3.6: Data-Flow Diagram Level-1 Institution

- **Institute Dashboard:** This is a special dashboard for the institute. It may provide features for managing the forum, such as moderating discussions and managing users.
- **Logins:** This process handles user authentication. When users provide their credentials, this process verifies them and grants access based on their roles.
- **Managing Events:** This process involves managing the events posted on the forum.
- **Accessing Forum Details:** This process involves accessing the details of the different discussion threads available in the forum.
- **Creating Forums:** Institution admins can create forums and their admins when a new forum is launched in their institute.

3.3 Database Design

The foundation of our forum system lies in a well-thought-out database schema, designed specifically for MongoDB. Instead of an Entity-Relationship (ER) Diagram, we utilize schemas to represent the structure of our data. These schemas serve as blueprints, outlining the organization of our database and the relationships between different data entities.

Schemas: Our MongoDB database consists of the following schemas:

- **chats:** Stores notification data from admins and sends them to users.

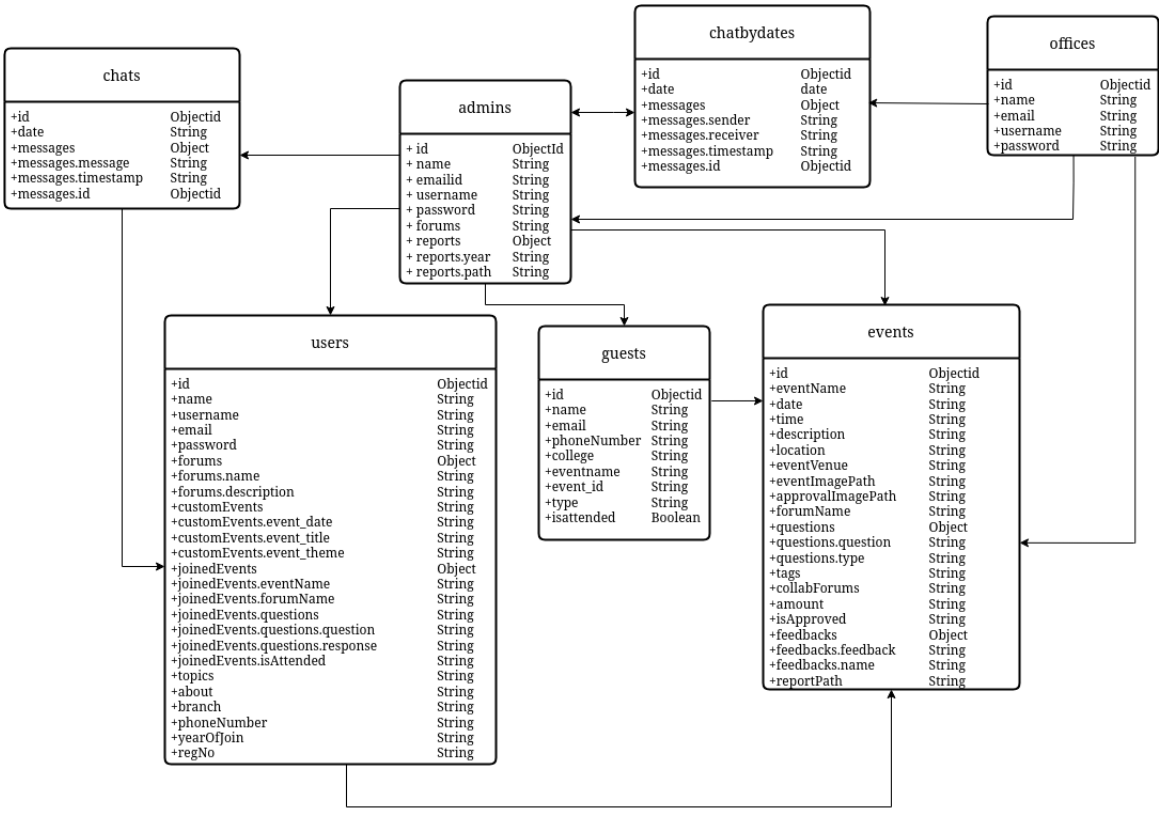


Figure 3.7: Schema Diagram

- **admins:** Stores details of executive committee (execom) admins, including their forum affiliations.
- **users:** Contains details of students, including events joined, topics of interest, and personal information.
- **guests:** For users outside the college who wish to join college events without registering.
- **offices:** Stores login details of institutional admins.
- **chatbydates:** Stores data exchanged between execom admins and institutional admins.
- **events:** Stores event details, feedback, reports, and registration fees.

While MongoDB doesn't strictly adhere to the traditional relational model, these schemas effectively organize our data and facilitate efficient access and retrieval.

3.4 Resource Requirements

This section delineates the hardware, software, and data prerequisites for the project working and system operation.

3.4.1 Hardware:

The minimum hardware configuration required for the proper functioning of the system can be outlined below:

- A Desktop or Laptop with i3 processor and 4GB Ram or above
- Internet Connectivity
- Hosting server-1 i.e. at College Advanced Machine Learning Lab, Room No. 206
- Hosting server-2 i.e. Created by one of the team member at his home.

3.4.2 Software:

The software we have used for development of the project has been outlined below.

- Operating System, Windows 10 or above, or the latest version of any Linux Distros.
- Development tools such as Visual Studio Code (VSCode) for coding and project management.
- Web Browsers like Google Chrome, Microsoft Edge, Mozilla Firefox
- Git and Github for Version Control
- Frontend development frameworks - NextJS, ReactJS
- Backend development framework, NodeJS with Express for web application development.
- Database management system - MongoDB.
- Payment Gateway: Stripe from Github Education Pack.
- API Key of Gemini 1 Pro model from Google[17] and Llama 3 from groq[16].
- Cloudflare, Nginx and Ngrok for configuration of server-2.
- OWASP ZAP[20] for Vulnerability Testing of the website.

3.4.3 Data Requirements:

To ensure the smooth functioning of the Forums Management System, we identified the following data requirements:

- Forum Details and Working: This includes comprehensive information about each forum, such as its purpose, objectives, activities, and events. We collaborated with the respective executive committee (execom) members of each forum to gather these details and formulate a structured plan for implementation.
- Logos of Forums: The logos representing each forum are essential for branding and visual identification on our website. We collected the logos of all forums to incorporate them into the user interface and enhance the overall user experience.

3.5 Server Configuration:

Our website is hosted on two servers. Server-1[15], provided in the College Lab, runs on Ubuntu and has a public IP address of 14.139.189.219. Server-2[14], created using an old laptop, also runs Ubuntu and was configured with the following technologies:

- Ngrok[4]: Ngrok serves as a pivotal component in our server setup, providing secure tunneling capabilities to enable external access while ensuring robust security measures are in place. It facilitates seamless and protected communication between our server and clients.
- Cloudflare[30]: As a trusted content delivery network (CDN) and domain name server provider, Cloudflare enhances the reliability and performance of our server infrastructure. By caching static content and optimizing domain resolution, Cloudflare minimizes latency and improves overall user experience. Additionally, its security features safeguard against online threats, ensuring data integrity and user privacy.
- Nginx[3]: Nginx functions as our primary web server and reverse proxy, efficiently managing incoming HTTP requests and optimizing resource allocation for enhanced performance. Its lightweight yet robust architecture enables swift content delivery and allows for seamless integration of security measures such as SSL/TLS encryption.
- Certbot Let's Encrypt[27]: Certbot Let's Encrypt plays a crucial role in securing our website by generating SSL certificates, enabling us to implement HTTPS encryption. This ensures that data exchanged between our server and clients is protected from unauthorized access and interception, bolstering overall security posture.

We have utilized an alternative cloud server through Azure [8], leveraging Microsoft's scalable and reliable cloud platform. This ensures optimal performance and accessibility for our web application. Azure's advanced infrastructure and comprehensive suite of services guarantee high availability and responsiveness, even under varying traffic conditions.

3.6 Algorithms

The project contains various algorithms for implementing its working including authentication, recommendation algorithm and others listed below.

3.6.1 Authentication Algorithm

This algorithm describes a two-factor authentication process for a user login system. It includes steps for both existing users and new users, with OTP verification for added security. For existing users, it validates credentials and sends an OTP to the registered email. For new users, it initiates a registration process, again with OTP verification.

Algorithm 1 Authentication Algorithm

```
Start
User authentication page loads
if user exists then
    User enters the username and password values in the login
    Credentials are validated with database
    if credentials are correct then
        6-digit OTP is sent to the email address corresponding to the user credentials
        Entered OTP is verified
        if OTP is correct then
            User dashboard is loaded
        else
            Authentication failure
        end if
    else
        Authentication failure
    end if
else
    Sign-up page is loaded for user registration with details Name, Email Address, Username, and Password
    6-digit OTP is sent to the email address entered
    if OTP entered by user is correct then
        Entered OTP is verified, and credentials are successfully entered to the database
    else
        Registration failure
    end if
end if
Stop
```

3.6.2 Adding Organisations (User) Algorithm

This algorithm describes the process of a user joining an organization within a system. It involves the user logging in, selecting an organization to join, and entering their membership ID. The system then verifies the user's details and membership ID with the organization's data. If the details are correct, the user is added to the organization; otherwise, the process fails.

Algorithm 2 Adding Organization to User Algorithm

```
Start
User logs in to the dashboard
When Join Organization is clicked, asks for the organization to join
if organization is selected then
    Asks to enter the membership ID of user
    Name of user, organization name and ID is verified with the JSON file in the server
    if the ID is correct corresponding to the user then
        The membership is added to the user
    else
        User is not part of the organization (Join Organization failure)
    end if
end if
Stop
```

3.6.3 Event Recommendation Algorithm

This algorithm outlines the steps for recommending events to a user based on their interests using a trained Doc2Vec model.

Algorithm 3 Event Recommendation Algorithm

```
Input: training.json (a JSON file containing user and event data)
Output: Recommended Events
Start
Train the Doc2Vec model:
    a. Load user and event data from training.json.
    b. Preprocess the data by tokenizing and creating TaggedDocuments for users and events.
    c. Train a Doc2Vec model using the preprocessed data.
Save the trained model to "eventmodel".
Retrieve the user with the given username.
    a. If no user is found, print an error message and exit.
Calculate the user's interest vector using the trained Doc2Vec model.
Find events similar to the user's interests:
    a. Calculate the similarity between the user's interests and each event's tags using Cosine similarity.
    b. Filter events based on a threshold similarity value and future event dates.
    c. Select the top recommended events.
Print the recommended events.
Save the recommended events to a JSON file with the format "usernameevents.json".
Stop
```

3.6.4 Organisation Event Creation

This algorithm outlines the process of creating an event in an organization's system. It involves the admin logging in, entering event details, and the system checking for scheduling conflicts before

either creating the event or prompting for a change in date or time.

Algorithm 4 Organization Event Creation

1: **Start**
2: The administrator logs in to the dashboard.
3: Upon selecting "Create Event," the system prompts the administrator to input event details, including name, date, time, and event poster. Once completed, proceed to step 4.
4: Prior to event creation, the system verifies the backend to ascertain if an event already exists at the specified date and time. If no conflicting event is found, proceed to step 5; otherwise, proceed to step 6.
5: The new event, along with its details and image, is transmitted to the backend server for storage. Subsequently, the database is updated, and the event date is marked on the calendar.
6: A warning message is displayed indicating, "An event is already scheduled for the specified date and time." The administrator is advised to select an alternative date or time.
7: **Stop**

3.6.5 Event Report Generation Algorithm

This algorithm describes the process of generating a report for an event by an admin. It involves selecting an event, automatically filling in details from the backend, entering the report body, and using a script to generate the report in a predefined template. The generated report can then be downloaded in PDF format.

Algorithm 5 Event Report Generation and Upload

1: **Start**
2: Admins initiate the process.
3: Admins are provided with the option to generate reports for events affiliated with their respective organizations.
4: Admins are granted access to a text editor interface to manually compose the report content.
5: Alternatively, admins have the option to utilize prompt text for report content generation, leveraging Generative AI technology.
6: Upon completion of report generation, admins have the capability to save the document as a PDF file.
7: Following PDF generation, admins proceed to submit the report via the designated "Upload Reports" feature, accessible within the Annual or Event Reports section.
8: **Stop**

3.6.6 Admin Chat

This algorithm outlines the steps for the Admin Chat functionality, including fetching sender and receiver IDs, loading previous chats, sending messages, and transmitting messages in real-time using Socket.IO.

Algorithm 6 Admin Chat

1: Start
2: Fetch sender and receiver IDs from the backend during page loading.
3: Load previous chats from the backend using the sender and receiver IDs.
4: When the user types a message and clicks on send, go to step 5.
5: Save the message in the backend with sender, receiver, and timestamp information.
6: Transmit the message through Socket.IO to the receiver in real-time.
7: Stop

Chapter 4

Report of Project Implementation

This chapter provides a comprehensive overview of the actualization of the project, detailing the execution of planned strategies, challenges encountered, and solutions implemented. It encompasses the deployment of software components, integration of features, and the achievement of project milestones.

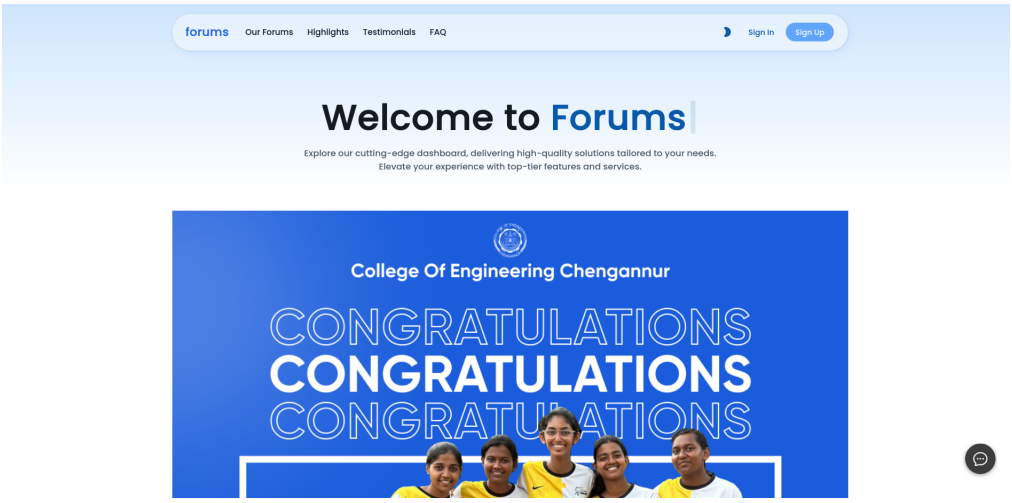


Figure 4.1: Landing Page of FMS

4.1 Landing Page & Guest View Module

The Landing Page is the first page we land on when we open the site. In our project, the landing page contains a gallery showcasing achievements related to forums, and a list of forums with their logos. These logos redirect to the respective sites of each forum. There is also a login button for students and admins to redirect to their respective dashboards. The main objective of the Landing Page is to provide an overall guest view about the forums in our campus and also provided an external join event feature for guests i.e. Participants out of the Institute.

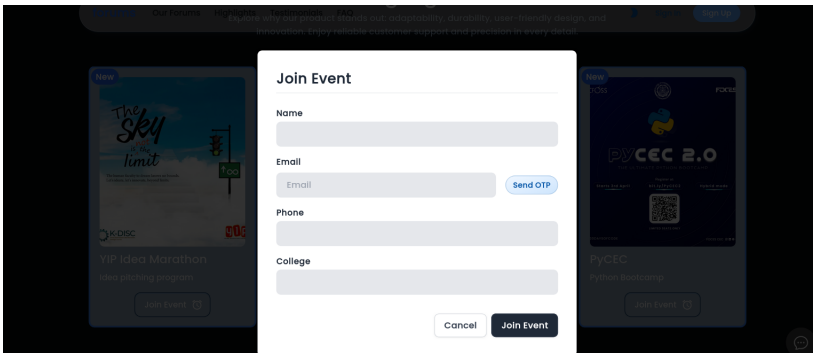


Figure 4.2: Guest Event Registration

4.2 Authentication Module:

The Authentication Module includes the Login and Sign-up page. It also includes two-factor authentication using OTP, which will be sent to their Email ID. Only students with the ceconline.edu domain can sign up and login to our website. We have included a Forgot Password facility, which is also handled using OTP. Additionally, for OTP generation and authentication, we utilize the EmailJS[18] service.

4.2.1 User Login

Username and password[21] is entered by the user. Then the credentials are verified with backend whether this user exists or not.If not, shows no user exists.Else, sends the OTP to the email address associated with that user. After successful OTP verification,user gets redirected to the home page. Admins have seperate page where the schema is different.

4.2.2 Forgot Password

User has to enter the email address to which OTP is to be sent for verification.It checks whether any user exists with this email address or not.If there is user,then OTP is sent to the user and after successful verification,it prompts to enter new password and after that,password gets updated successfully and redirects to login.

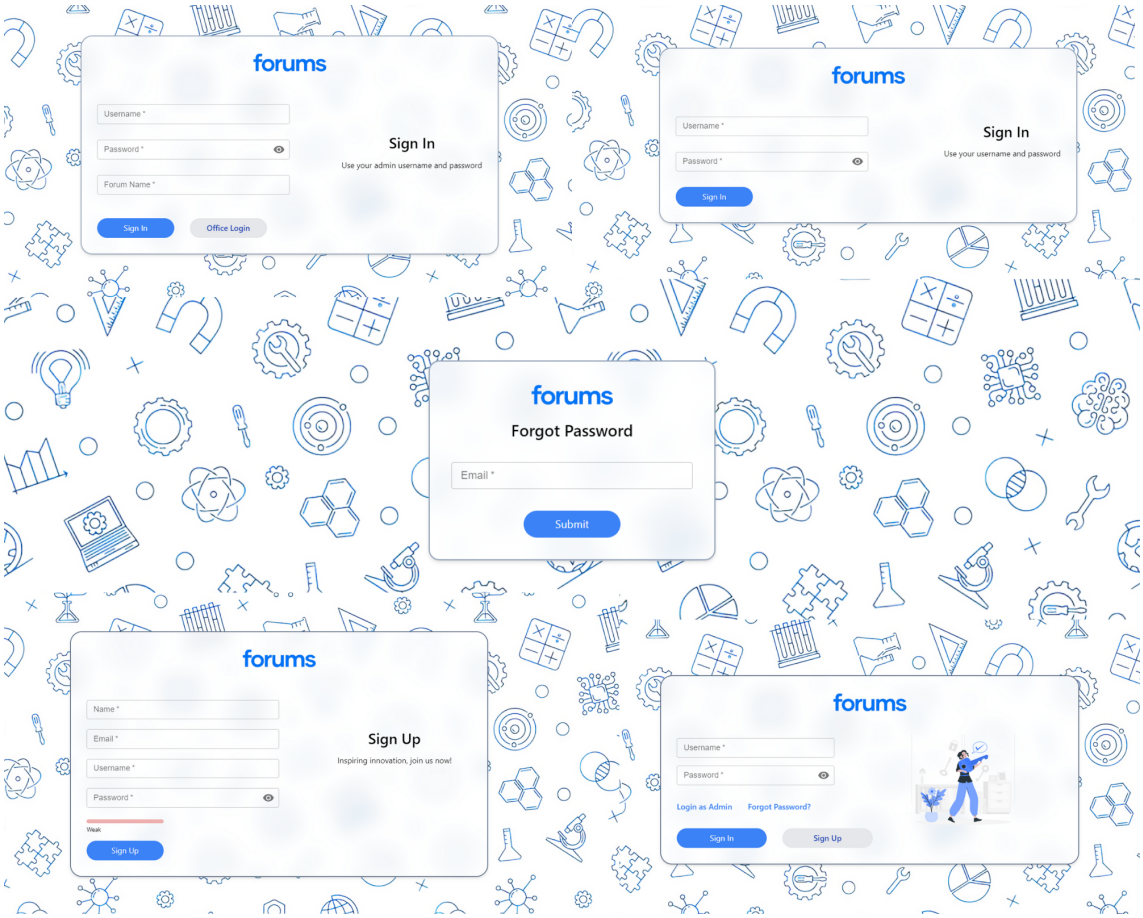


Figure 4.3: Authentication Modules

4.2.3 Sign Up

User enter details like Name,email,username and password.Then it checks whether an user already exists or not.If not,sends 6 digit OTP to email using emailjs service.Then after successful verification,account gets created successfully and redirects to login.

4.3 Admin Module

The Admin Module is designed for the Executive Committee of each forum. Each forum is provided with one admin account to access the site. It includes features for admins to create and post events, view organization members, and send notification messages to community members. Additionally, it offers an analytics page, calendars, and attendance tracking functionalities.

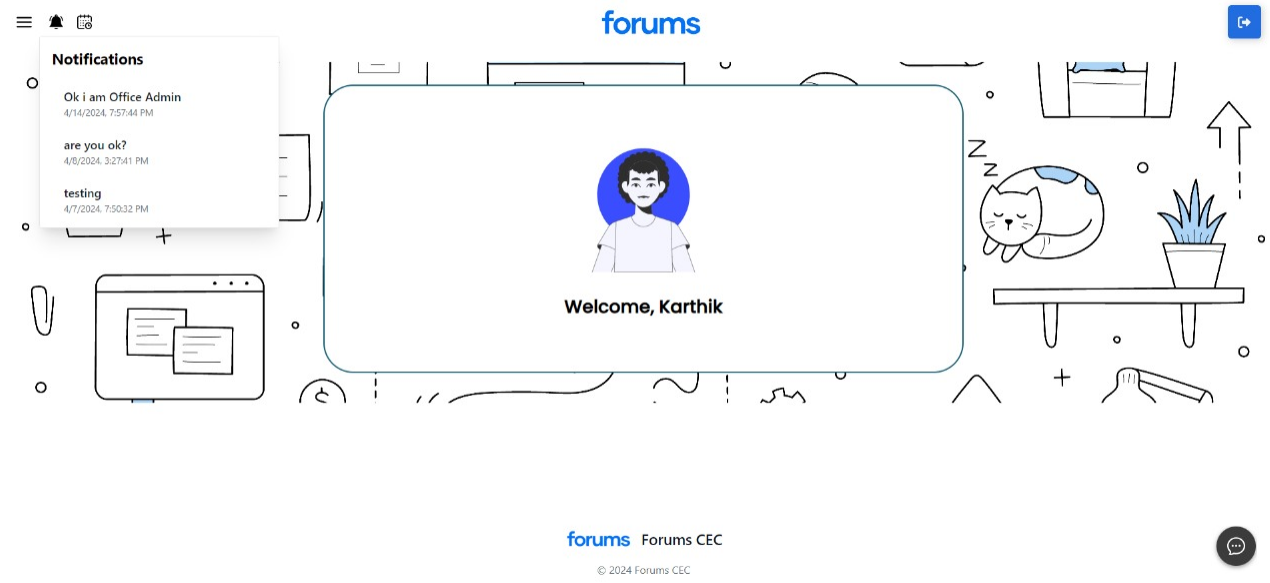


Figure 4.4: Admin Dashboard

4.3.1 Calendar

The Calendar feature in the Community Admin Module assists admins in identifying events hosted by their own and other forums. Admins can schedule their next event conveniently based on available dates.

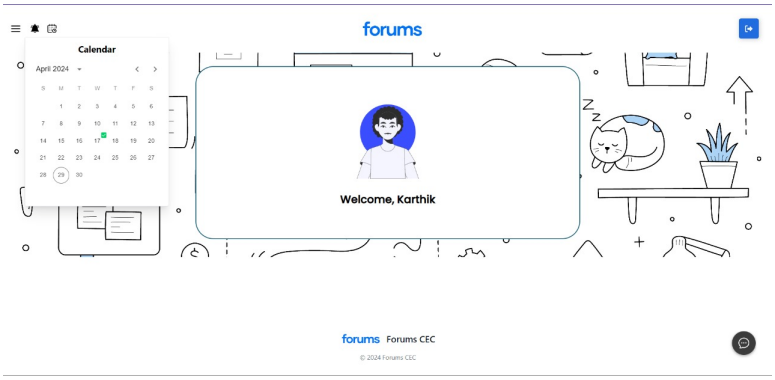


Figure 4.5: Admin Calendar

4.3.2 Members List

The Members section displays a list of members logged into the specific forum along with their details. Admins can generate a list of members in CSV format and upload member details using the import function.

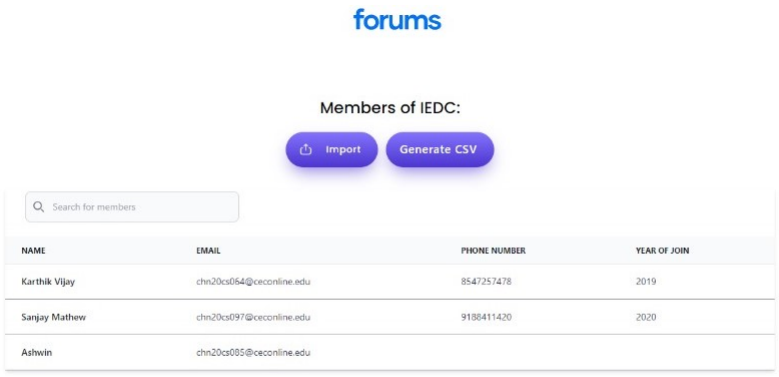


Figure 4.6: Admin Members List

4.3.3 Events

The Events section allows admins to create and manage events for the specific forum.

Event Creation

Admins can create events by entering details such as name, description, amount (if applicable), time, location, and poster. Additionally, they must upload a digital letter signed by their Community's faculty advisor.

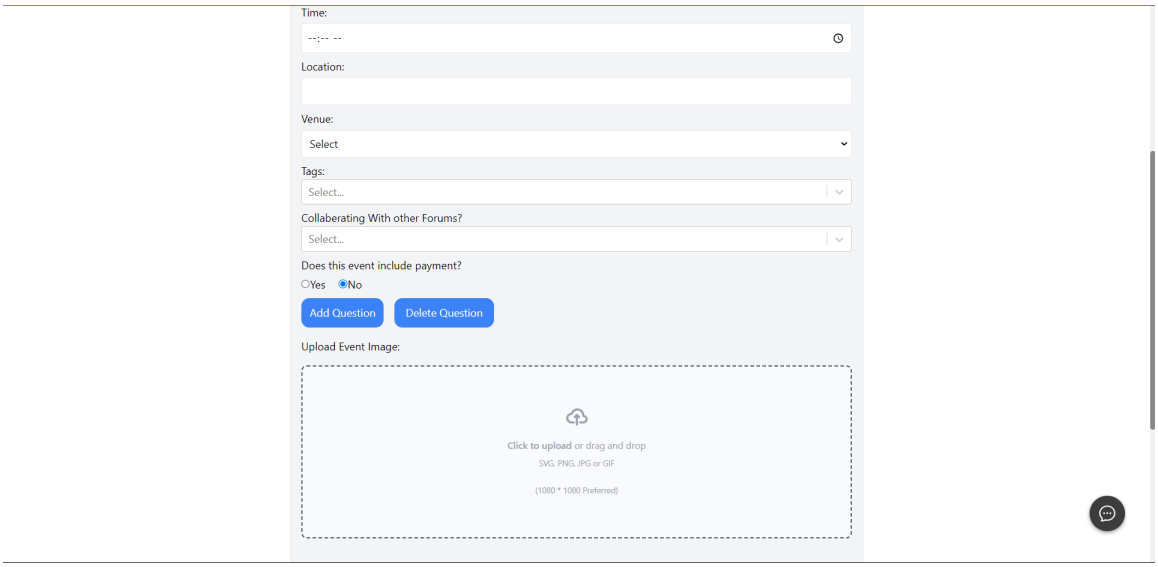


Figure 4.7: Create Event: Admin

Event List

Admins can view a list of events they have created.

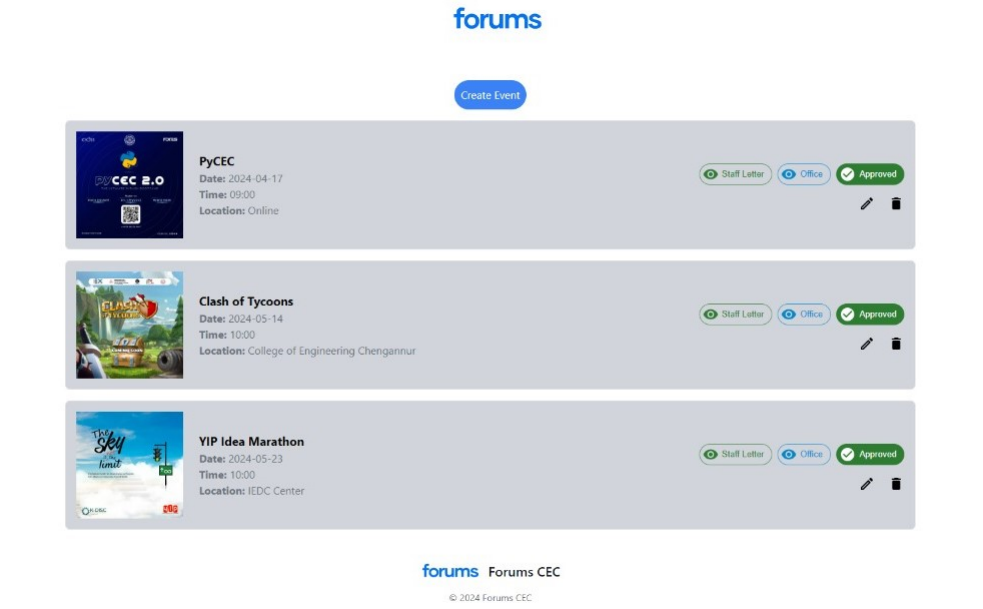


Figure 4.8: Admin Event List

Event Description

Admins can view the description of specific events as entered during their creation.

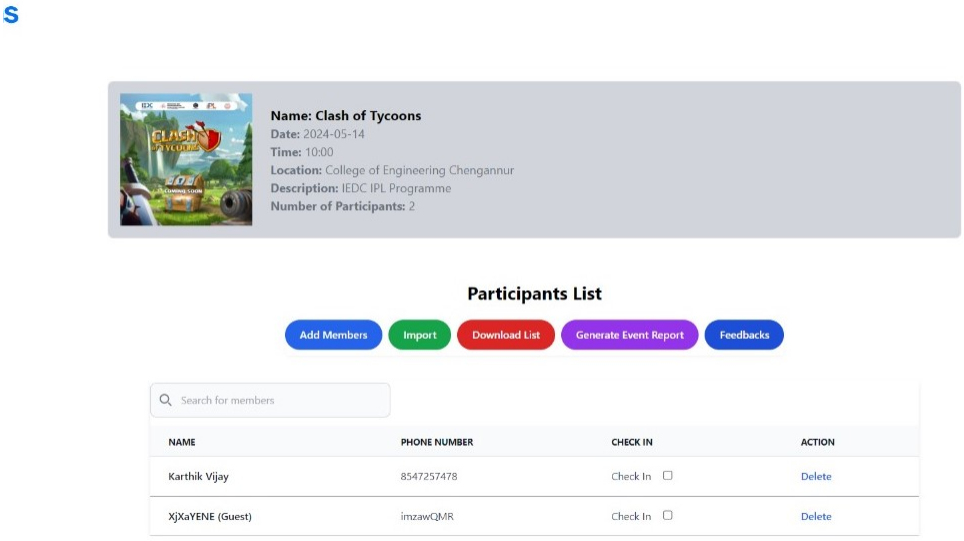


Figure 4.9: Event Description: Admin

Report Generation aided by AI

Here, Execom Admins can create their reports and generate PDFs. They can input event details and utilize generative AI models such as Gemini and Llama 3 to generate the report content.

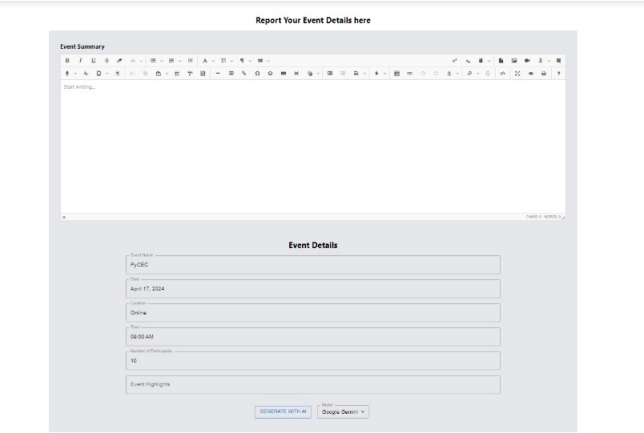


Figure 4.10: Report Generation Page

Description of Gemini and llama 3:

- **Gemini:** Utilizes Gemini 1 Pro[17] from Google's maker suite API key. Gemini is a powerful AI tool capable of generating text based on prompts provided by the user. It leverages advanced natural language processing (NLP) techniques to produce coherent and contextually relevant text.
- **llama 3:** Developed by Meta, llama 3 utilizes the Groq[16] service for generative AI capabilities. Similar to Gemini, llama 3 accepts prompts from users and generates text based on the input. It's designed to understand and mimic human language patterns, producing output that aligns with the provided prompts.

These AI models enhance the report generation process by automating content creation based on user prompts, thereby streamlining the workflow for Execom Admins.

4.3.4 Analytics

The Analytics page provides comprehensive insights through graphs and charts[23], enabling Community Admins to make informed decisions regarding event planning and management.

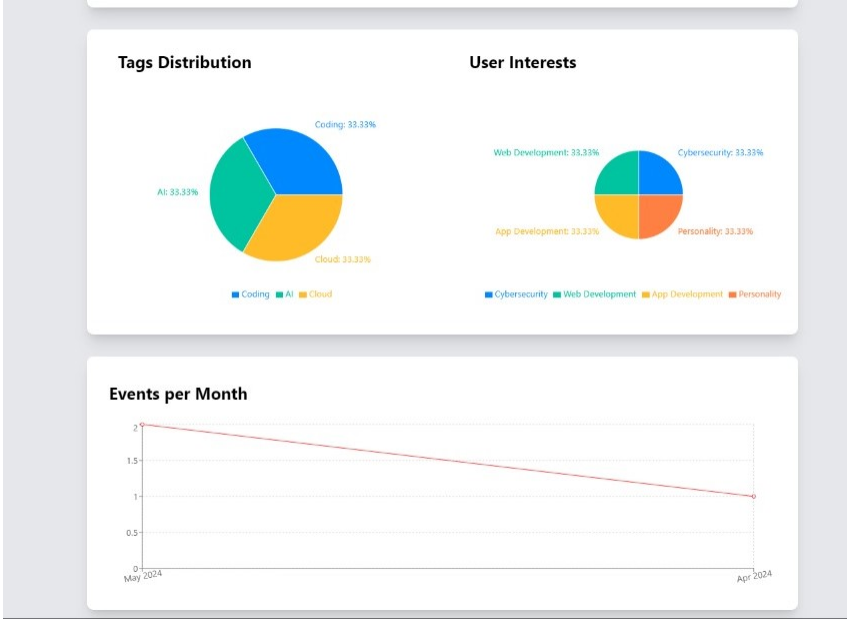


Figure 4.11: Forums Analytics

The analytics include:

- **Events Distribution by Month:** Visual representation of event distribution over different months, facilitating trend analysis and strategic scheduling.
- **Tags Distribution:** Pie chart depicting the distribution of tags across events, aiding in identifying popular topics and themes among community members.
- **User Interests Charts:** Pie charts displaying user interests based on their interactions and preferences, empowering admins to tailor events according to community interests and preferences.

This analytical data serves as a valuable resource for Community Admins, enabling them to strategically plan and host events in alignment with prevailing trends and community preferences.

4.3.5 Connect With Institutional Admins

This feature enables quick communication between Institutional admins from both sides, facilitating seamless collaboration. It is implemented using Socket.IO for real-time messaging.

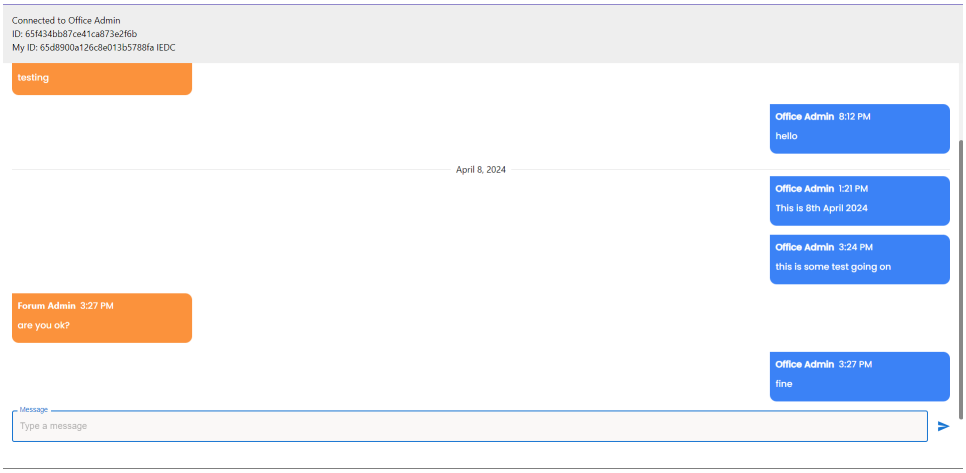


Figure 4.12: Connect Admins

4.3.6 Upload Report

Community admins can upload their event reports and annual reports through this feature. Institutional admins can access these reports as needed.

4.4 Institute Admin

The Institute Admin module is designed for College/Institutional authorities such as the Principal, Managers, and Superintendents. They can monitor the functioning of Forums and have the authority to approve or reject events. Additionally, they can communicate easily with Community Admins via a socket communication chat.

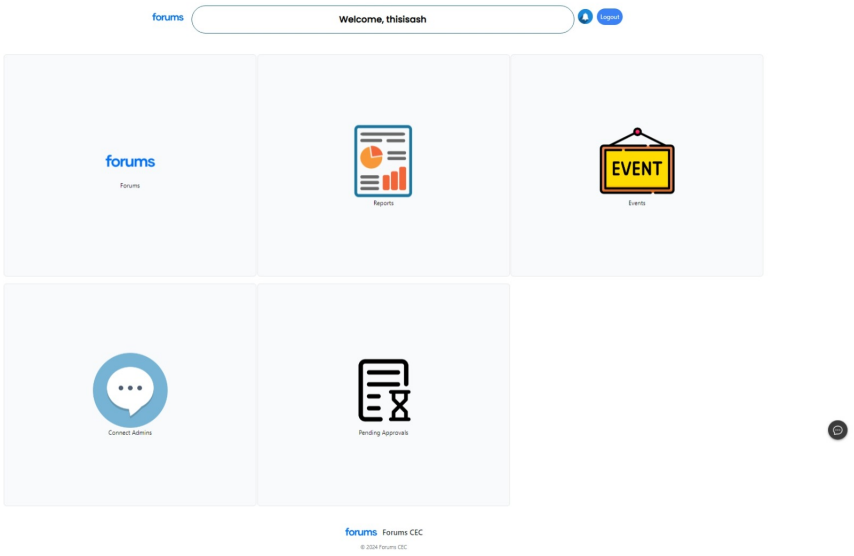


Figure 4.13: Institute Admin Dashboard

4.4.1 Forums

Institutional Admins can view the forums in their institution and create new forum accounts to provide access to new communities within their institution.

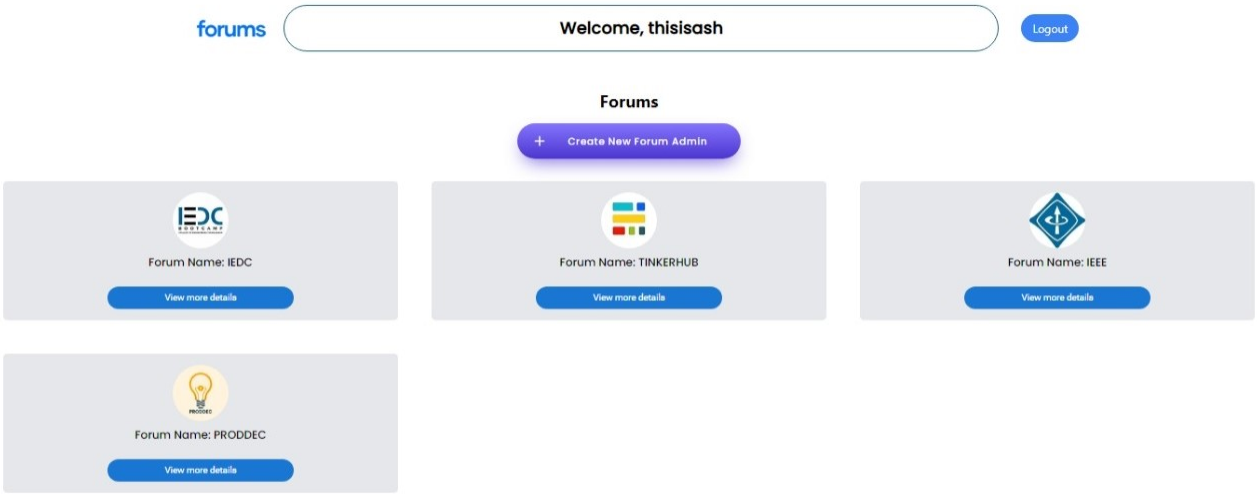


Figure 4.14: Forums List: Institute Admin

4.4.2 Events List

Institutional Admins can view the events hosted at their institution by every forum here.

4.4.3 Connect Admins

Institutional Admins can connect with Community Admins and send messages using socket communication.

4.4.4 Event Approval System

After Community Admins create an event, it is sent here for Institutional Admins to approve. They can reject it if there are any issues and view the letter from the community's advisor. Additionally, they can send digital approval back to Community Admins.

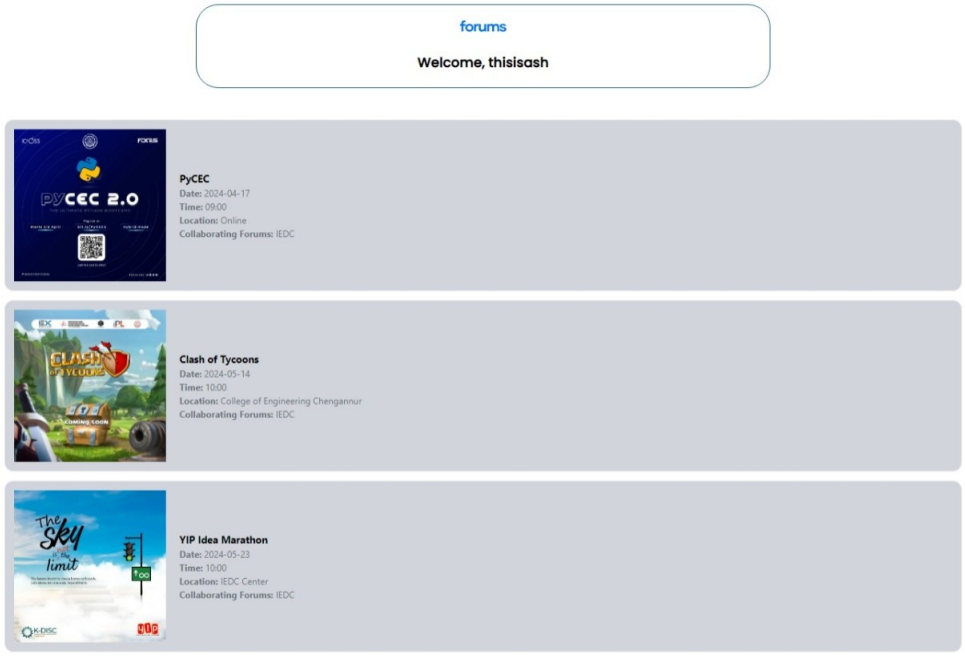


Figure 4.15: Events List: Institute Admin

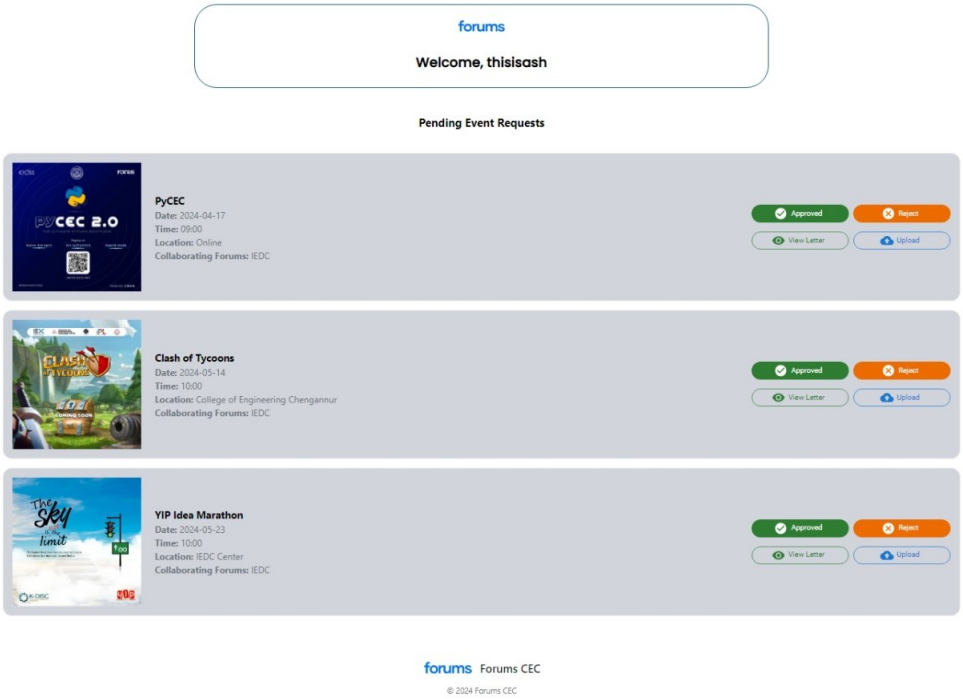


Figure 4.16: Event Approval: Institute Admin

4.4.5 Upload Reports

Institutional Admins can access reports uploaded by Community Admins for each community, both annual and specific event-based.

4.5 Student Module:

The Student Module includes the Dashboard for student members of the campus to engage themselves with forums. Here, the student can view their participation in events of each forum. Other features that will be implemented in the coming days include easy registration for events and an upcoming events gallery.

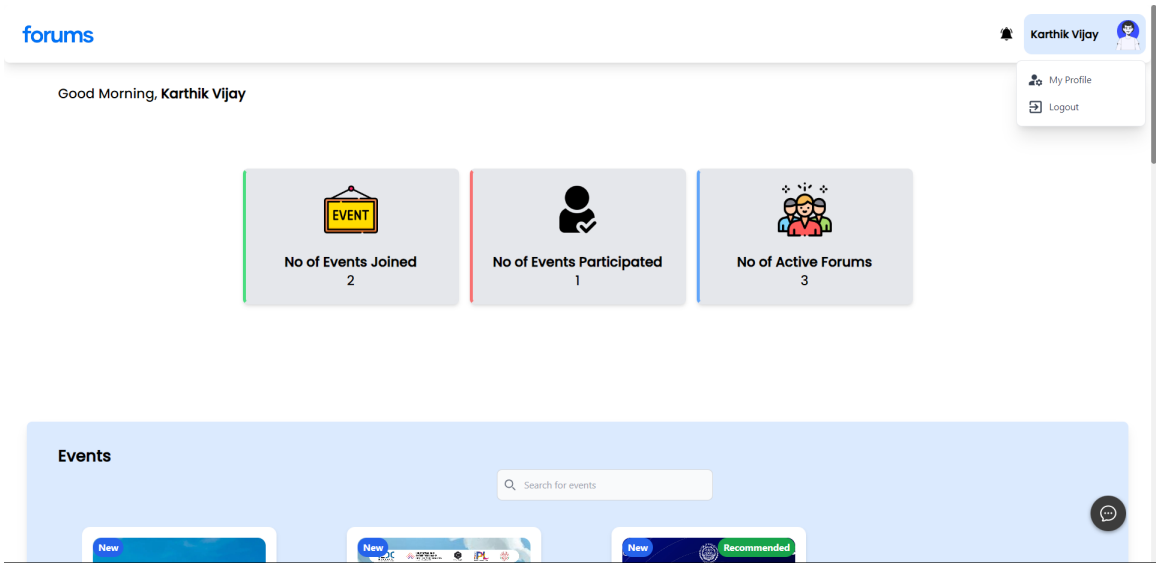


Figure 4.17: Dashboard: Student User

4.5.1 Event Recommendation

The Event Recommendation feature enriches user engagement by offering personalized event suggestions based on students' domains and interests, designated with the "Recommended" tag.

To accomplish this, we employ Gensim[29], a widely-used Python library for natural language processing tasks including topic modeling and document similarity analysis. Specifically, we utilize the Doc2Vec algorithm, a variant of the Word2Vec model tailored for documents. Doc2Vec represents documents as continuous vectors in a high-dimensional space, capturing semantic similarities between them.

The process commences with tokenizing and preprocessing the text data, which involves removing stop-words and numeric characters to extract meaningful features. Subsequently, the Doc2Vec model is trained on a corpus consisting of user profiles and event tags. During training, the model learns to generate vector representations for each document, thereby capturing the underlying semantic relationships between topics and events.

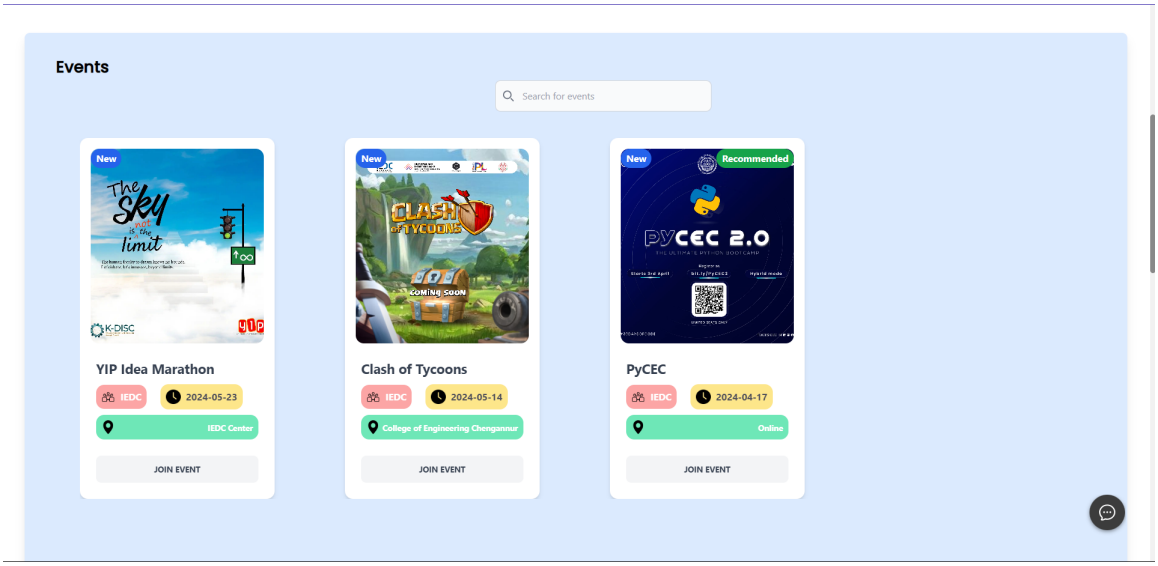


Figure 4.18: Event Recommendation

By training the model on this amalgamated corpus, it acquires the ability to discern the semantic representations of both user profiles and event tags. Consequently, it can efficiently generate recommendations by computing the similarity between a user's profile and event tags. This facilitates the identification of pertinent events based on the user's interests and preferences.

The recommendation system enhances user engagement and satisfaction by furnishing personalized event suggestions tailored to each user's individual interests and domain expertise. This contributes to a more enriching and tailored experience for users interacting with the event management platform.

4.5.2 Payment Gateway - Stripe API

[12] The transaction process for student event registration within the forums management system entails secure payment handling facilitated by the integrated Stripe API. This integration ensures seamless and reliable transactions for event registration, enhancing the overall user experience.

We obtained access to the Stripe API through the GitHub Student Developer Pack, which provides students with access to valuable tools and services. The Stripe API enables secure debit and credit card payments, allowing students to conveniently complete event registrations without compromising their financial security.

To integrate Stripe into our website, we utilized the Stripe.js package, which provides a straightforward and efficient means of incorporating Stripe's payment processing functionality. This package allows us to seamlessly embed payment forms into our website, enabling students to securely enter their payment details and complete transactions with ease.

By leveraging the Stripe API, we ensure that payment processing is handled securely and efficiently, safeguarding users' sensitive financial information throughout the transaction process. This not

only instills confidence in users but also streamlines the event registration process, contributing to a seamless and hassle-free user experience.

Overall, the integration of the Stripe API into our forums management system enables secure and reliable payment processing for event registration, enhancing the functionality and usability of the platform for students.

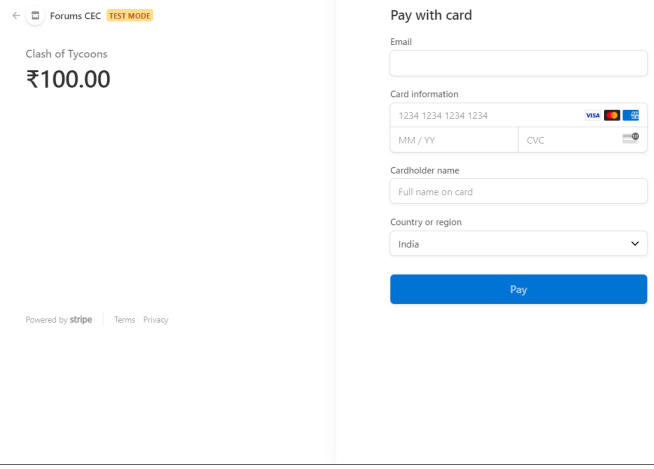


Figure 4.19: Payment Gateway

4.6 Chatbot for Help

In addition to the existing functionalities of our forums management system, we have implemented a Chatbot powered by Botpress[28]. This Chatbot serves as a valuable tool for assisting users and providing them with relevant information and support.

The Chatbot is trained with data extracted from our website, enabling it to familiarize itself with the various aspects and features of the platform. This knowledge base equips the Chatbot with the necessary information to effectively respond to user queries and provide assistance as needed.

To integrate the Chatbot into our website, we utilize the Botpress framework, a powerful platform for building and deploying conversational AI applications. This framework offers a range of features and capabilities for developing intelligent chatbots with natural language understanding and context-awareness.

Through the integration process, the Chatbot is seamlessly embedded into the user interface of our website. Users can easily access the Chatbot from any page of the website, allowing them to engage in conversations and seek assistance in real-time.

The Chatbot enhances the user experience by providing a convenient and accessible channel for obtaining information and support. Whether users have questions about the platform's features, need assistance with navigation, or require troubleshooting help, the Chatbot is available to assist them every step of the way.

By leveraging the capabilities of Botpress, we ensure that the Chatbot delivers accurate and helpful responses to user inquiries, contributing to a more efficient and satisfactory user experience within our forums management system.

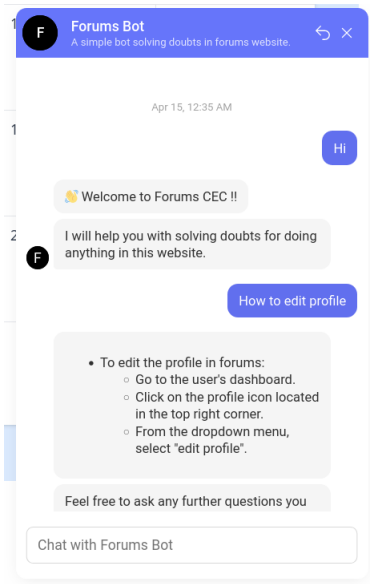


Figure 4.20: Chatbot

Chapter 5

Security Testing

The security of our web application hosted in server-2 was rigorously tested using OWASP ZAP[20] (Open Web Application Security Project Zed Attack Proxy), which is an open-source web application security scanner used to identify security vulnerabilities in web applications. It offers automated scanners for common security issues like cross-site scripting (XSS), SQL injection, and insecure configurations. Additionally, ZAP provides tools for manual testing and analysis, enabling users to inspect HTTP requests and responses, simulate attacks, and analyze application behavior. It's widely utilized by developers, security professionals, and organizations to ensure the security of their web applications.

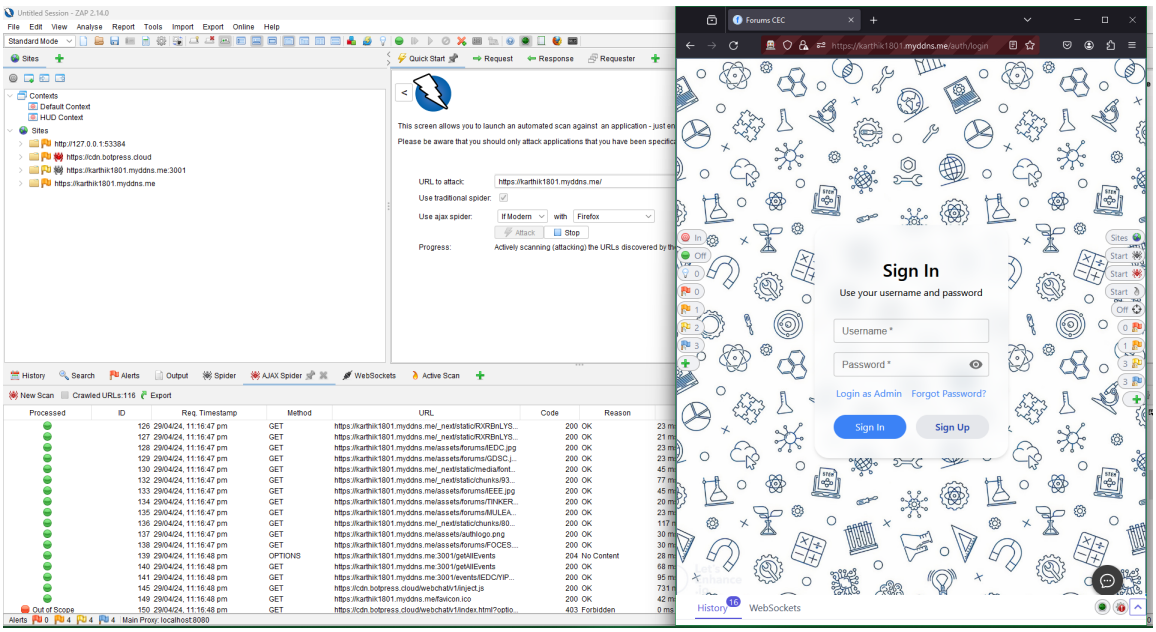


Figure 5.1: Chatbot

The following table presents the findings of the security testing, including the number of alerts by alert type and their associated risk levels:

(The percentages in brackets represent each count as a percentage, rounded to one decimal place,

of the total number of alerts included in this report.)

Alert type	Risk	Count
CSP: Wildcard Directive	Medium	1 (8.3%)
Content Security Policy (CSP) Header Not Set	Medium	4 (33.3%)
Cross-Domain Misconfiguration	Medium	1 (8.3%)
Missing Anti-clickjacking Header	Medium	1 (8.3%)
Server Leaks Version Information via "Server" Header	Low	12 (100.0%)
Strict-Transport-Security Header Not Set	Low	69 (575.0%)
Timestamp Disclosure - Unix	Low	10 (83.3%)
X-Content-Type-Options Header Missing	Low	66 (550.0%)
Information Disclosure - Suspicious Comments	Informational	25 (208.3%)
Modern Web Application	Informational	4 (33.3%)
Re-examine Cache-control Directives	Informational	5 (41.7%)
Retrieved from Cache	Informational	12 (100.0%)
Total		**12**

Alert Descriptions with Solutions

- **CSP: Wildcard Directive (Medium):**
  - **Alert Description:** Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.
  - **Solution:** Addressed CSP (Content Security Policy) alert by configuring the application to set a strict CSP header.
- **Content Security Policy (CSP) Header Not Set (Medium):**
  - **Alert Description:** Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.
  - **Solution:** [Ensure that your web server, application server, load balancer, etc. is configured to set the Content-Security-Policy header.

    3. Content Security Policy (CSP) Header Not Set (Medium):
- **Cross-Domain Misconfiguration (Medium):**
  - **Alert Description:** [The response does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.]

  - **Solution:** [Modern web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. Ensure one of them is set on all web pages returned by your site/app.]

- **Server Leaks Version Information via "Server" Header (Low):**
  - **Alert Description:** The web/application server is leaking version information via the "Server" HTTP response header. Access to such information may facilitate attackers identifying other vulnerabilities your web/application server is subject to.
  - **Solution:** Ensure that your web server, application server, load balancer, etc. is configured to suppress the "Server" header or provide generic details.

- **Strict-Transport-Security Header Not Set (Low):**
  - **Alert Description:** HTTP Strict Transport Security (HSTS) is a web security policy mechanism whereby a web server declares that complying user agents (such as a web browser) are to interact with it using only secure HTTPS connections (i.e. HTTP layered over TLS/SSL).
  - **Solution:** Ensure that your web server, application server, load balancer, etc. is configured to enforce Strict-Transport-Security.

- **X-Content-Type-Options Header Missing (Low):**
  - **Alert Description:** The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type.
  - **Solution:** Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.

- **Timestamp Disclosure - Unix (Low):**
  - **Alert Description:** A timestamp was disclosed by the application/web server - Unix.
  - **Solution:** Manually confirm that the timestamp data is not sensitive, and that the data cannot be aggregated to disclose exploitable patterns.

- **Retrieved from Cache (Informational):**
  - **Alert Description:** The content was retrieved from a shared cache. If the response data is sensitive, personal or user-specific, this may result in sensitive information being leaked.
  - **Solution:** Validate that the response does not contain sensitive, personal, or user-specific information. If it does, consider using HTTP response headers like Cache-Control, Pragma, and Expires to prevent caching.

Additional Security Measures Implemented:

Implemented HTTPS[27] protocol to bolster the security of our home server (server-2) infrastructure. Leveraged nginx in conjunction with ngrok to secure access to the domain[14]. Utilized Certbot Let's Encrypt to generate SSL/TLS certificates, ensuring robust encryption and authentication. Furthermore, orchestrated the migration of the HTTP website to HTTPS, enhancing data integrity and confidentiality.

Chapter 6

Results & Conclusions

This chapter summarizes the outcomes of the project, highlighting the achievements, benefits, and reflections on the technical skills acquired during implementation. It also discusses the transformative impact of technology on campus forums and outlines future opportunities for innovation.

6.1 Results

The implemented system offers a centralized platform, simplifying forum management for administrators and providing students with unified access to all forums. This consolidation facilitates seamless participation in various events and activities.

We integrated a recommendation algorithm to personalize event suggestions based on individual student interests, past participation, forum activity, and preferences. The algorithm refines suggestions using machine learning techniques. Additionally, a feedback mechanism allows students to provide ratings and reviews, enhancing real-time insights. Administrators can effortlessly generate comprehensive reports with the rich text editor feature, further augmented by artificial intelligence for content insights and formatting suggestions.

Overall, these enhancements aim to elevate the forum management system, providing a dynamic and engaging platform that aligns closely with the preferences of the student body.

6.2 Benefits

The project yields numerous benefits. Administrators now enjoy a unified platform for managing forums efficiently, streamlining their tasks. Simultaneously, students benefit from a single point of access, simplifying their engagement with forums, events, and activities across the campus. This centralized approach fosters a more connected and engaged campus community.

6.3 Technical Knowledge Gained & Concluding Thoughts

The development journey significantly expanded our technical prowess. We delved into the intricacies of the Next.js framework, React, Node.js, Express.js, MongoDB, and EmailJS authentication.

Additionally, we conducted web penetration testing to ensure the security and integrity of the system.

In conclusion, this project not only provided a practical application of these technologies but also elevated our understanding and proficiency in web development. The Forums Management System stands as a testament to our acquired skills, offering a valuable solution that enhances the organizational and participatory aspects of campus life. As we reflect on the journey, we recognize the transformative impact of technology on campus forums, bringing a new level of efficiency and engagement to our college community.

References

[1] OpenJS Foundation. "Node.js v21.4.0 documentation." Node.JS. https://nodejs.org/en/docs/. (accessed Apr. 11, 2024)

[2] Vercel. "NextJS Documentation." NextJS. https://vercel.com/docs. (accessed Dec. 11, 2023)

[3] Nginx. "Nginx Documentation." Nginx. https://nginx.org/en/docs/. (accessed Dec. 11, 2023)

[4] Ngrok. "Ngrok Documentation." Ngrok. https://ngrok.com/docs. (accessed Dec. 11, 2023)

[5] OpenJS Foundation. "ExpressJS 5.X API Reference" ExpressJS. http://expressjs.com/en/5x/api.html. (accessed Dec. 11, 2023)

[6] K. Vijay. "Event Listing Website." GitHub. https://github.com/Project-ASK/event-listing. (accessed Dec. 10, 2023)

[7] K. Vijay. "Event Lister." Event Lister. https://event-listing-ten.vercel.app/. (accessed Dec. 10, 2023)

[8] Microsoft. "Azure Documentation." Microsoft. https://learn.microsoft.com/en-us/azure/?product=popular. (accessed Dec. 5, 2023)

[9] Yepdesk. "Yepdesk Event Management Website." Yepdesk. https://www.yepdesk.com/. (accessed Dec. 1, 2023)

[10] Konfhub. "Konfub Event Management Website." Konfhub. https://konfhub.com/. (accessed Dec. 1, 2023)

[11] Google & Bevy. "GDSC." Google Developer Student Club https://gdsc.community.dev/. (accessed Mar. 29, 2024)

[12] Stripe. "Stripe Payment Gateway" Stripe https://docs.stripe.com/. (accessed April 10, 2024)

[13] npm. "Jodit." Jodit https://docs.stripe.com/. (accessed April 4 2024)

[14] K. Vijay, "Forums." Forums CEC http://karthik1801.myddns.me/. (accessesed April 30, 2024)

[15] College Of Engineering Chengannur, "Forums." Forums CEC http://14.139.189.219/. (accessed April 30, 2024)

[16] groq. "Llama 3 Model API." groq. https://console.groq.com/docs/quickstart. (accessed April 30, 2024)

[17] Google. "Gemini Pro 1 API." Google AI Studio. https://aistudio.google.com/app/apikey. (accessed April 20, 2024)

[18] EmailJs. "Email JS authentication" EmailJS. https://www.emailjs.com/docs/.(accessed March 10, 2024)

[19] MongoDB and JavaScript. "Mongoose JS" MongooseJS https://mongoosejs.com/docs/. (accessed April 25, 2024)

[20] OWASP. "Zed Attack Proxy." ZAP. https://www.zaproxy.org/. (accessed April 27th, 2024)

[21] npm. "bcryptjs." npm. https://www.npmjs.com/package/bcryptjs. (accessed March 4, 2024)

[22] npm. "file-saver." npm. https://www.npmjs.com/package/file-saver. (accessed March 29, 2024)

[23] recharts. "react charts."recharts. https://recharts.org/en-US/. (accessed April 14,2024)

[24] socket.io. "Socket.io-client." socket.io. https://socket.io/docs/v4/. (accessed April 19, 2024)

[25] Material UI. "Material UI." mui. https://mui.com/material-ui/getting-started/. (accessed April 29, 2024)

[26] Flowbite. "Flowbite." Flowbite. https://flowbite.com/docs/getting-started/introduction/ (accessed April 15, 2024)

[27] Let's Encypt "HTTPs Certification" Let's Encrypt https://letsencrypt.org/getting-started/. (accessed April 28, 2024)

[28] Botpress. "Botpress Chatbot" Botpress. https://botpress.com/docs/. (accessed April 29, 2024)

[29] Radimrehurek, "Doc2Vec." gensim. https://radimrehurek.com/gensim/apiref.html. (accessed April 15, 2024)

[30] Cloudflare Inc. "Cloudflare" Cloudflare. https://www.cloudflare.com/. (accessed March 20, 2024)