Forums Management System

CSD416 Project Phase II

Interim Report - February 2024

CS20C48 CHN20CS097 Sanjay Mathew
CS20C43 CHN20CS085 R Ashwin
CS20C031 CHN20CS064 Karthik Vijay
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

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

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

- **Forum 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

2.1 Literature Review

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 Forum 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 Forum 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 endeavor, 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 Forum 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), and “Backend Frameworks” (the specific technologies used to build the backend).
- **Database:** This is where data is stored and managed.

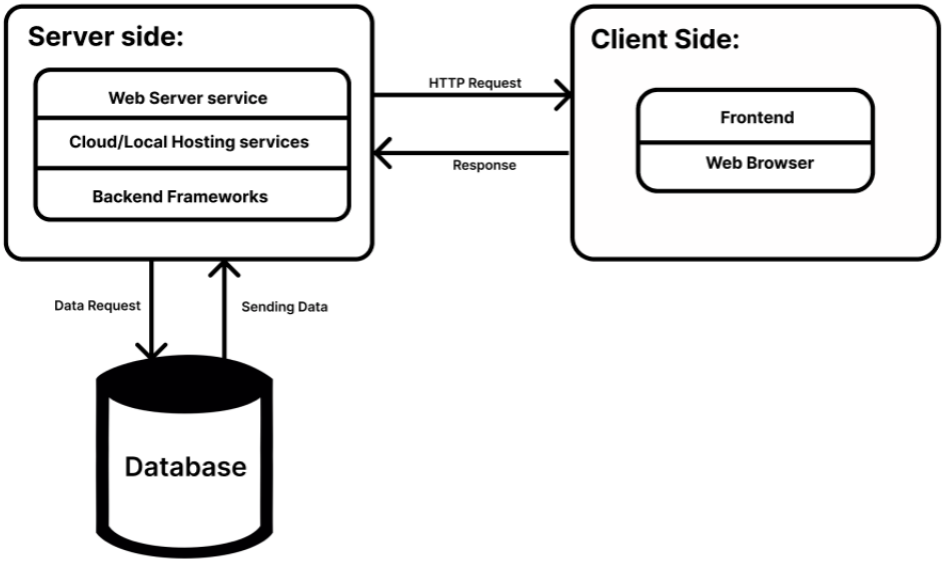


Figure 3.1: Web Application Architecture: General Design

3.1.2 Web Application Architecture: Initial Implementation Design

End Users:

- Institutional Admins: Responsible for overseeing and managing the overall functioning of the Forum Management System.
- Forum Execom: 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 used to build the user interface, ensuring a responsive and dynamic experience. [2]

Server Side:

- Ngrok: Utilized for secure and efficient tunneling, enabling secure access to our server from external environments.[4]
- Azure: The web application is hosted on Microsoft Azure, providing a scalable and reliable cloud platform to ensure optimal performance and accessibility.[8]

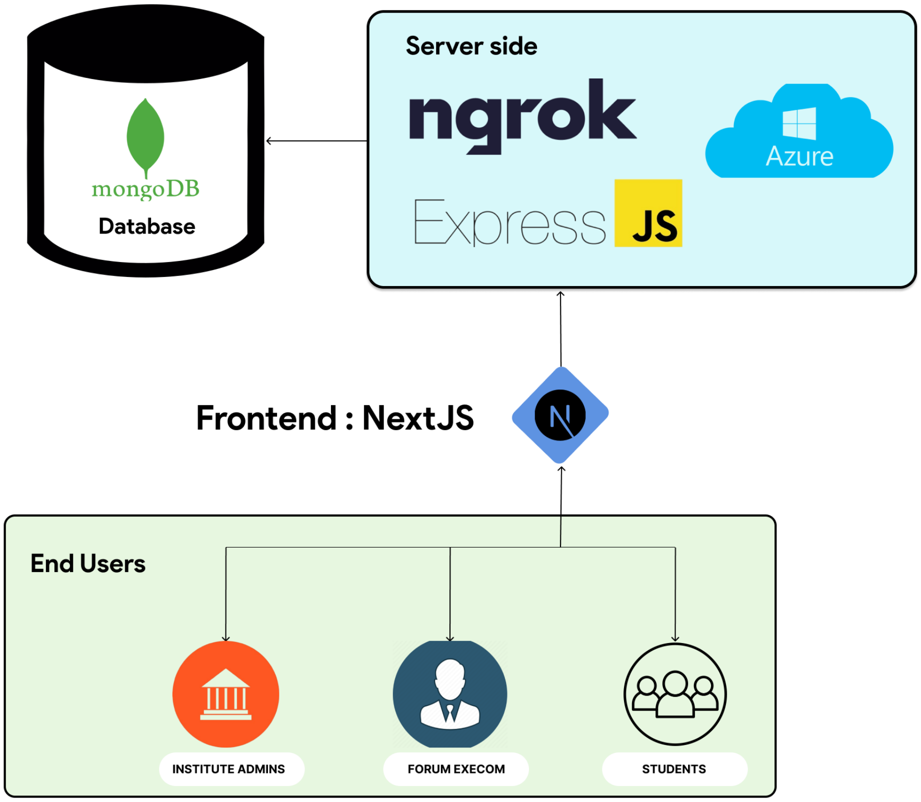


Figure 3.2: Web Application Architecture: Initial Implementation Design

- Express.js: Chosen as the server-side framework, Express.js facilitates the development of robust and efficient server-side applications.[5]

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.

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.

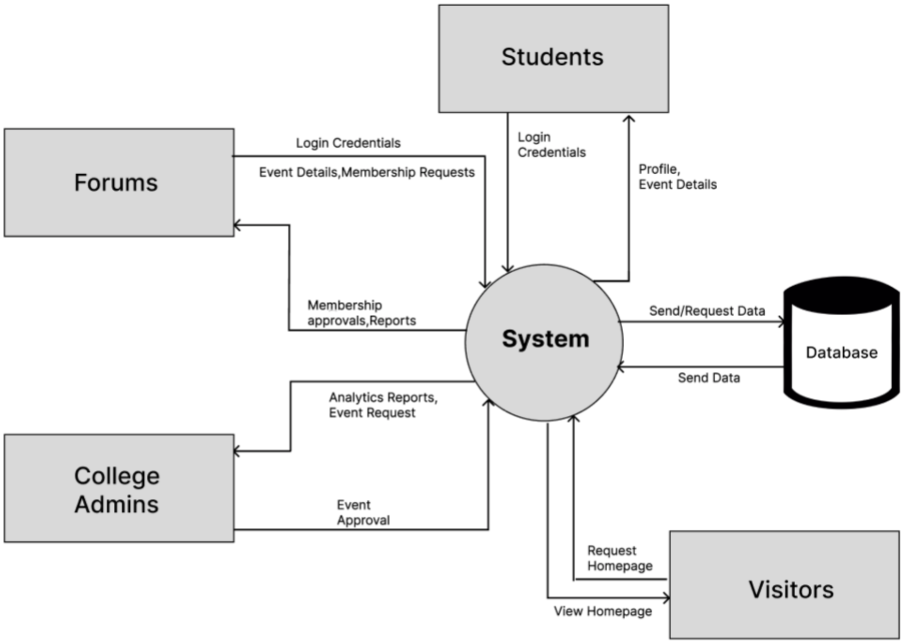


Figure 3.3: Data-Flow Diagram Level-0

- 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.

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.

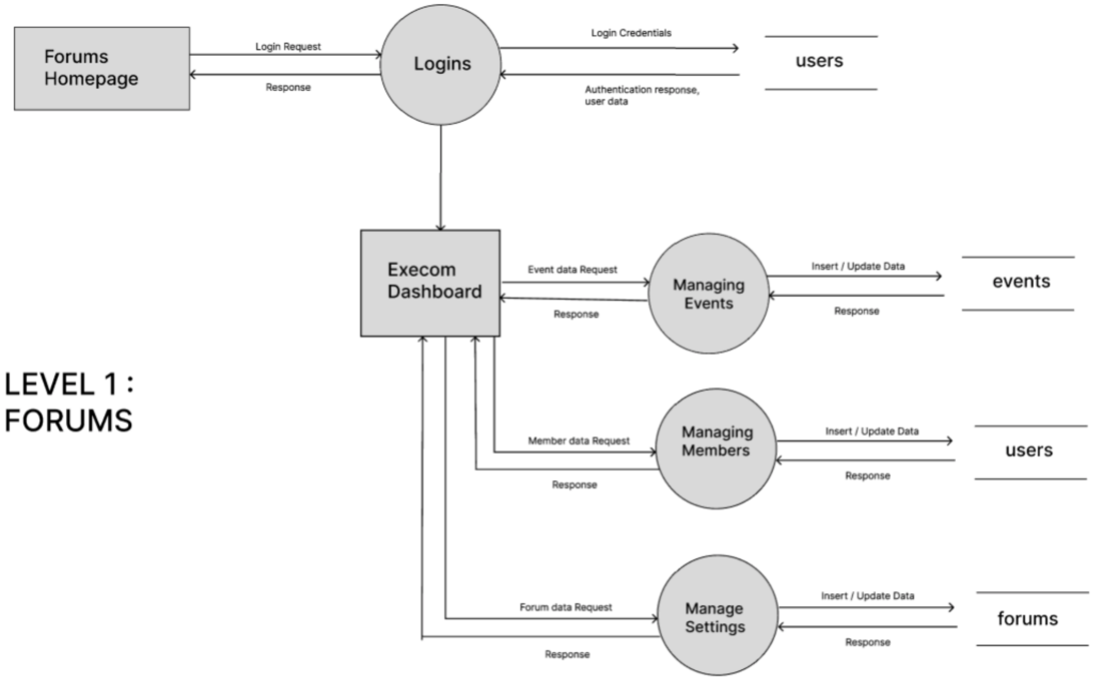


Figure 3.4: Data-Flow Diagram Level-1 Forums

Forums:

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.
- **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). 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.

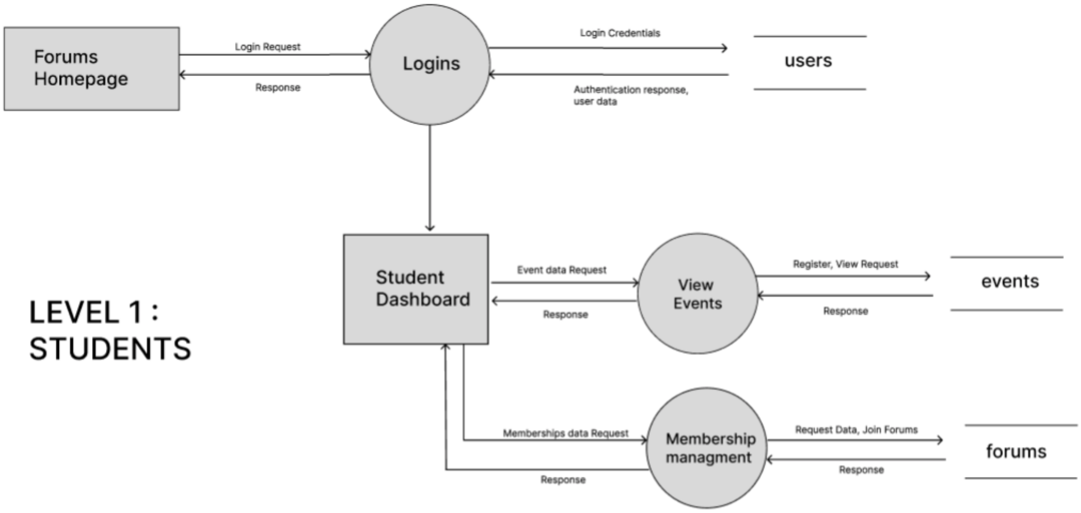


Figure 3.5: Data-Flow Diagram Level-1 Students

- **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.
- **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 Students Details:** This process involves accessing the details of the students of the institution.
- **Accessing Forum Details:** This process involves accessing the details of the different discussion threads available in the forum.

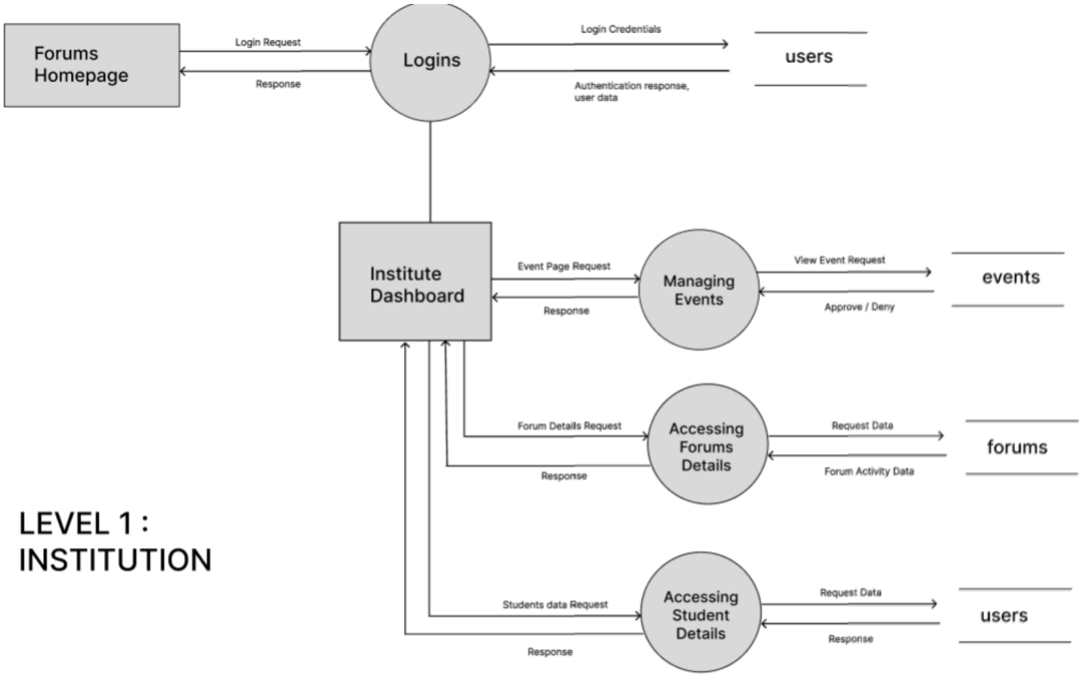


Figure 3.6: Data-Flow Diagram Level-1 Institution

3.3 Database Design

The foundation of our forum system lies in a well-thought-out database design, visualized through an Entity-Relationship (ER) Diagram. This diagram serves as a crucial guide, offering a comprehensive overview of the data entities and relationships within our system. It acts as a blueprint, illuminating the structure of our database and facilitating a clear understanding of data flow and dependencies.

Entities: These represent the fundamental building blocks of our system, each playing a distinctive role:

- **Users:** Individuals registered within the forum, engaging in various activities such as content creation, discussions, and event participation.
- **Events:** Diverse events posted on the forum, each associated with a user who acts as the event creator.
- **Members:** Registered users with additional privileges, including forum participation and event engagement.

Relationships: These connections define the interplay between entities, elucidating the dynamics of our system:

- **Schedules:** Event schedules providing structured timelines, each linked to a specific event.

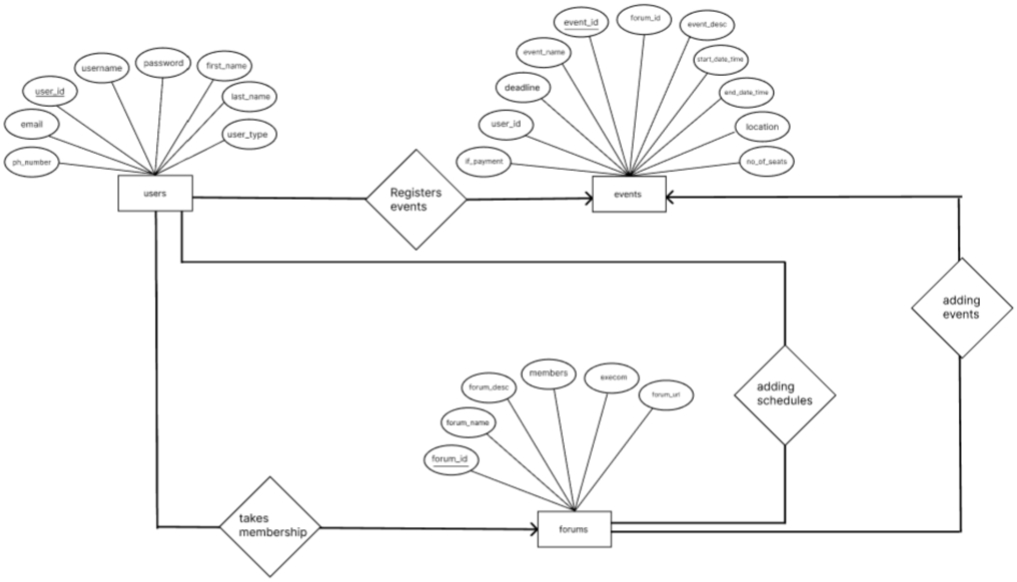


Figure 3.7: ER Diagram

- **Adding Events:** Action representing users adding new events to the forum.
- **Adding Schedules:** Action representing users adding schedules for events.
- **Registers Events:** Action representing users registering for various events.

3.4 Hardware & Software Requirements

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 i.e. at College Advanced Machine Learning Lab, Room No. 206

3.4.2 Software:

- 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.

3.5 Algorithms

3.5.1 Authtentication 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.

```
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
```

3.5.2 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 1 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
```

3.5.3 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 2 Organization Event Creation

```
1: Start
2: Admin logs in to the dashboard.
3: When "Create Event" is clicked, the system prompts for the event details, including name,
   date, time, and event poster. After filling in the details, proceed to step 4.
4: Before creating the event, the system checks the backend to see if any event already exists on
   the entered date and time. If no event exists, go to step 5; otherwise, go to step 6.
5: The new event is sent to the backend server along with the event image and details. The
   database is updated, and the calendar date is marked with the event.
6: A warning message is displayed, stating "An event is already scheduled on that particular date
   and time." The admin is instructed to choose a different date or time.
7: Stop
```

3.5.4 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 3 Event Report Generation

```
Start
Admins can generate the report of events for each event in their corresponding organization.
Admins can select the event name for which report is to be generated.
Event Name, Organization name, and other details are filled out automatically by fetching from backend.
Enter the body of the report by the admin.
When submitted, using the report generator script, it collects the data entered and fills in the predefined report template.
After successfully generating the report, admins can download the report in PDF format.
Stop
```

3.6 Work Schedule

3.6.1 Works Done: Progress Overview

- **August - September 2023:** In this initial phase, our focus will be on project planning, team formation, and assessing the availability of necessary resources. We will ensure that all required hardware and software resources are in place, and any deficiencies will be addressed promptly.

- **October - December 2023:** These months will be dedicated to system design and architecture development. Our team will collaboratively design the structure and core functionality of the web application, outlining its key components and user interfaces.

- **January - February 2024:** In these months, we implemented the module of authentication completely, while implementing the Student and Admin modules partially. We have also configured our site in a server we assembled in the College lab, and the website is publicly accessible.

3.6.2 Work Schedule: Next Steps

As we transition into the interim phase, we have reached the halfway point in the project implementation.

- **March 2024:** Completion of the remaining modules for both admin and user.

- **April 2024:** Testing of the website, along with checks and improvements to the backend.

Chapter 4

Report of Project Implementation

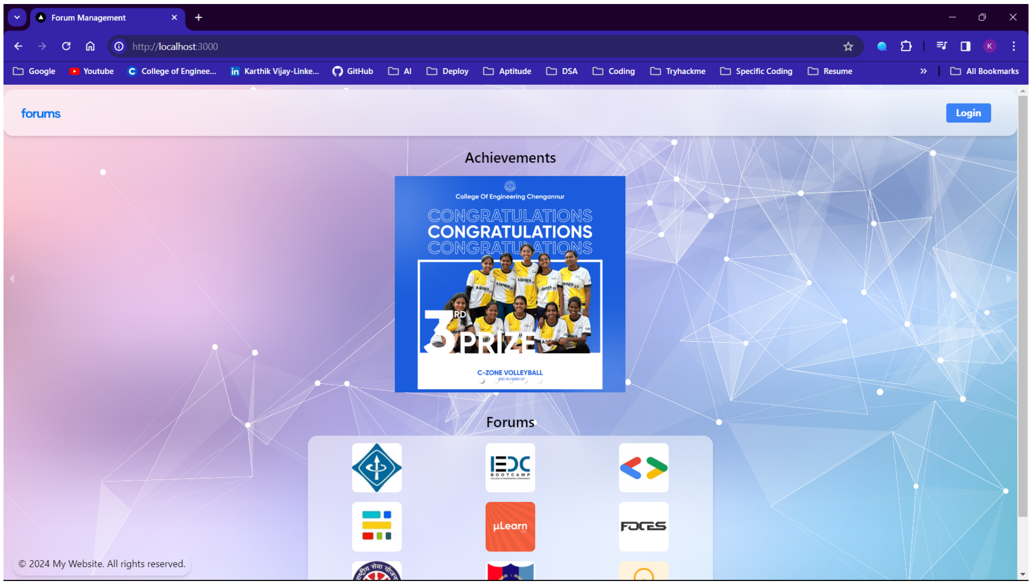


Figure 4.1: Landing Page of FMS

4.1 Landing Page Module

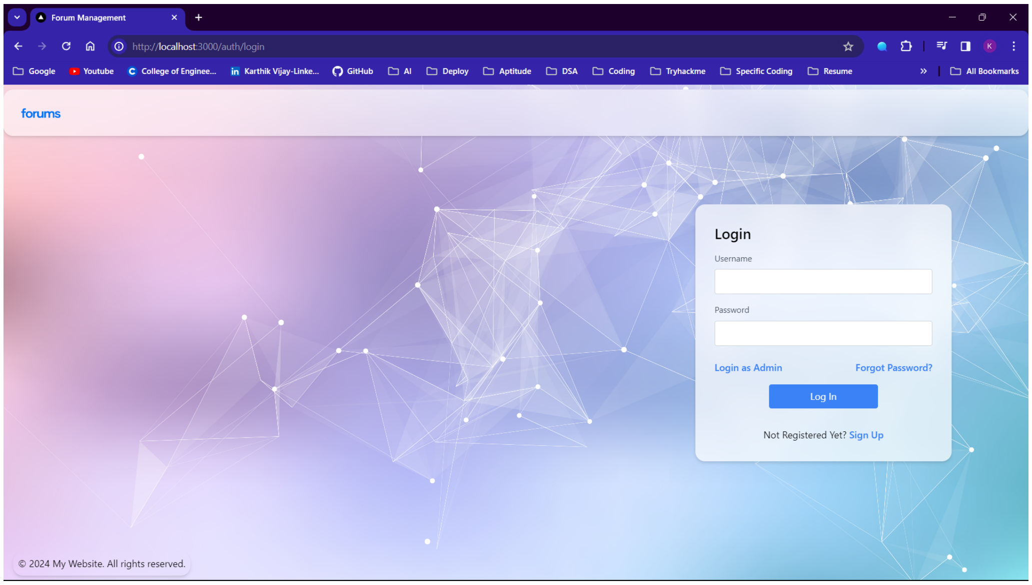
The Landing Page is the first page we land on when we open the site. In our project, the landing page contains a gallery showcasing achievements related to forums, and a list of forums with their logos. These logos redirect to the respective sites of each forum. There is also a login button for students 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.

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.

4.2.1 User Login

Username and password 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.



© 2024 My Website. All rights reserved.

Figure 4.2: Login Page: User

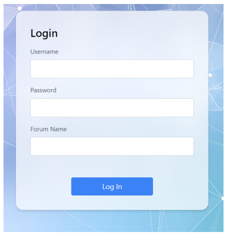


Figure 4.3: Login Page : Admin

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

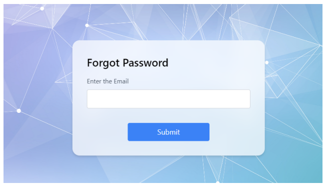


Figure 4.4: Forgot Password

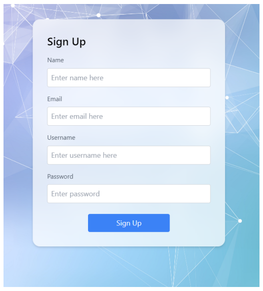


Figure 4.5: Sign Up Page

4.2.3 Sign Up

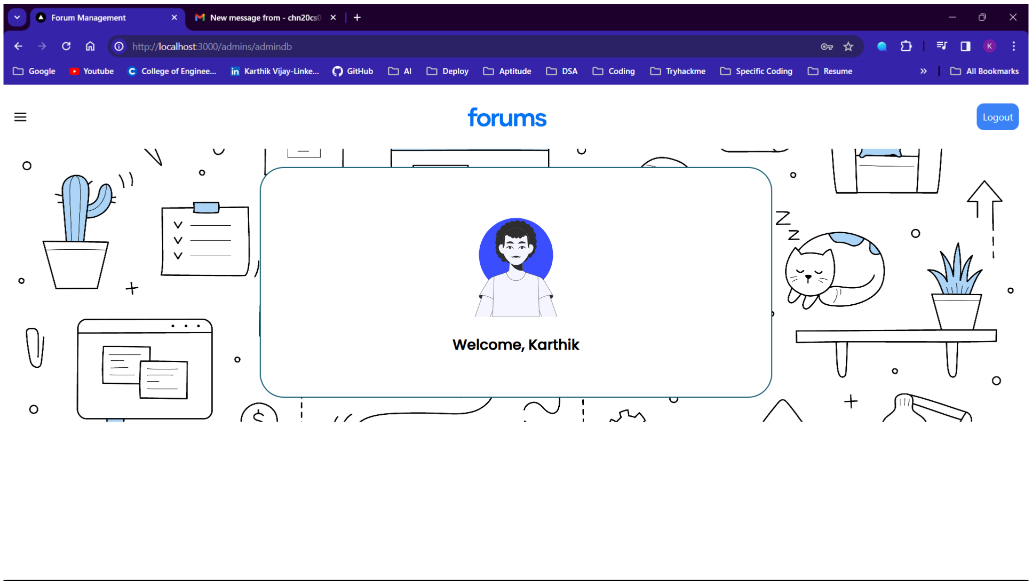
User enter details like Name,email,username and password.Then it checks whether an user already exists or not.If not,sends 6 digit OTP to email using emailjs service.Then after successful verification,account gets created successfully and redirects to login.

4.3 Admin Module:

The Admin Module is for the Executive Committee of each forum. Each forum will be given one account for accessing the site as an admin. This module is under development, where we have added features allowing admins to create and post their events and view their Organization members. Features which are still in development include an analytics page, mass-emailing, calendars, and attendance tracking.

4.4 Student Module:

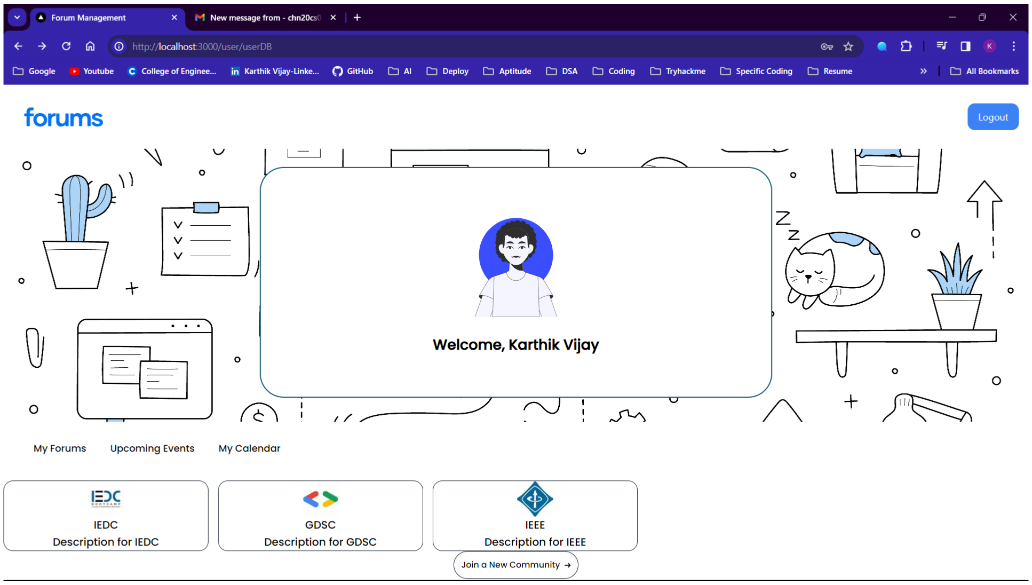
The Student Module includes the Dashboard for student members of the campus to engage themselves with forums. Here, the student can view their participation in events of each forum. Other



Welcome, Karthik

Figure 4.6: Dashboard: Admin

features that will be implemented in the coming days include easy registration for events and an upcoming events gallery.



Welcome, Karthik Vijay

IEDC
Description for IEDC

GDSC
Description for GDSC

IEEE
Description for IEEE

Figure 4.7: Dashboard: Student User

Chapter 5

Results Conclusions

5.1 Project Overview

Our development efforts have successfully culminated in the creation of a robust Forums Management System tailored for managing multiple organization forums within the campus. The system is composed of four main modules: the Landing Page Module, Authentication Module, Admin Module, and Student Module. Each module plays a crucial role in providing a comprehensive solution for efficient forum management.

5.2 Results Proposed Plan for Future Enhancements

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

Looking forward, we envision enhancing the user experience by incorporating a recommendation algorithm. This algorithm will personalize event recommendations based on individual student interests, thereby increasing engagement and enriching the overall user experience. This proposed plan aims to elevate the forums system to new heights and align it more closely with the preferences of the student body.

5.3 Benefits

The benefits of our project are manifold. Admins 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.

5.4 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.

The configuration of servers and the process of hosting a web application publicly were also part of our learning journey.

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 Dec. 11, 2023)

[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 Dec. 11, 2023)