COLLEGE COMMUNITY APP

A PROJECT REPORT

Submitted by

ARULNITHI K	811722104015
DEEPAK MADHU KUMAR N	811722104026
HARIHARAN V M	811722104048

in partial fulfillment of the requirements for the award degree of

Bachelor in Engineering

20CS7503 - DESIGN PROJECT - 3

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(AUTONOMOUS)

SAMAYAPURAM - 621112

NOVEMBER 2025

K.RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS) SAMAYAPURAM - 621112

BONAFIDE CERTIFICATE

The work embodied in the present project report entitled "**COLLEGE COMMUNITY APP**" has been carried out by the students **ARULNITHI K, DEEPAK MADHU KUMAR N, HARIHARAN V M** The work reported herein is original and we declare that the project is their own work, except where specifically acknowledged, and has not been copied from other sources or been previously submitted for assessment.

Date of Viva Voice: ……………………

Mrs. S.Gayathri, M.E,	**Mr. R. Rajavarman, M.E.,(Ph.D.,)**
SUPERVISOR	HEAD OF THE DEPARTMENT
Assistant Professor	Assistant Professor
Department of CSE	Department of CSE
K Ramakrishnan College of Technology (Autonomous)	K Ramakrishnan College of Technology (Autonomous)
Samayapuram – 621 112	Samayapuram – 621 112

INTERNAL EXAMINER **EXTERNAL EXAMNIER**

ABSTRACT

The College Community App is a web-based social networking system designed to strengthen communication and collaboration within a campus environment. Built using Django, SQLite, and a responsive Bootstrap-driven frontend, the platform offers college students a secure digital space where they can interact, share updates, and stay informed about campus events. By blending modern backend logic with an intuitive UI, the application creates a cohesive space that reflects the real dynamics of an academic community. A key innovation in this system is its college-email-only authentication mechanism, which ensures that only verified students gain access. This selective approach builds trust, eliminates external interference, and promotes genuine participation within the platform. The system supports essential social features such as posts, media uploads, likes, comments, and customizable profiles, allowing students to express themselves and engage with peers in meaningful ways. Beyond social interaction, the app integrates a community events module that enables students to create, publish, and explore upcoming programs such as hackathons, cultural fests, and club activities. This transforms the platform into a centralized hub for both information sharing and student involvement. Overall, the project demonstrates a practical full-stack implementation.

Keywords: Django Framework, SQLite3, Student Networking, Campus Communication, User Interaction, Social Media Features, Event Calendar, Profile Management, Media Sharing, Secure Login, Web Application, Academic Community Platform.

ACKNOWLEDGEMENT

We thank our **Dr. N.Vasudevan**, Principal, for his valuable suggestions and support during the course of my research work.

We thank our **Mr.R. Rajavarman,** Head of the Department, Assistant Professor, **COMPUTER SCIENCE AND ENGINEERING**, for his valuable suggestions and support during the course of my research work.

We wish to record my deep sense of gratitude and profound thanks to my Guide **Mrs.S.Gayathri,** Assistant Professor, **COMPUTER SCIENCE AND ENGINEERING**, for her keen interest, inspiring guidance, constant encouragement with my work during all stages, to bring this thesis into fruition.

We are extremely indebted to our project coordinator **Mrs. R. Ramasaraswathi,** Assistant Professor, **COMPUTER SCIENCE AND ENGINEERING**, for her valuable suggestions and support during the course of my research work.

We also thank the faculty and non-teaching staff members of the **COMPUTER SCIENCE AND ENGINEERING**, **K RAMAKRISHNAN COLLEGE OF TECHNOLOGY**, Samayapuram, for their valuable support throughout the course of my research work.

Finally, we thank our parents, friends and our well wishes for their kind support.

SIGNATURE

____

____

____

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

CSS	-	Cascading Style Sheets
HTML	-	Hyper Text Markup Language
UI	-	User Interface
API	-	Application Programming Interface
HTTP	-	Hyper Text Transfer Protocol
UX	-	User Experience
2FA	-	Two-Factor Authentication

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The College Community App is a dedicated web-based platform developed to enhance communication, collaboration, and engagement within a college environment. It addresses the need for a unified digital space where students can connect, share information, and participate in campus activities without relying on external social networks. By integrating essential social interaction features with campus-specific functionalities, the system replicates the essence of college life in an accessible and structured digital form. The platform is designed to be intuitive, secure, and supportive of the everyday needs of students, ensuring that communication remains relevant and focused.

Technically, the application is built using Django as the core backend framework, supported by SQLite3 for data storage and Bootstrap, SCSS, and Django Template Language for the frontend interface. This full-stack combination enables efficient data management, responsive design, and seamless interaction between users and the system. The inclusion of features such as post sharing, profile customization, media uploads, and event creation demonstrates the practical implementation of CRUD operations, authentication workflows, and relational data handling.

A key aspect of the system is the college-email-only authentication mechanism, which ensures that access is restricted to verified students within the institution. This design choice strengthens the platform's privacy, fosters a trusted community environment, and helps maintain the authenticity of interactions. Additionally, modules such as the event calendar, user follow system, and search functionality contribute to a comprehensive ecosystem that supports both social and academic engagement.

1.2 PROBLEM STATEMENT

College students often rely on scattered digital platforms to communicate, share updates, and coordinate academic or extracurricular activities. Traditional notice boards, informal messaging groups, and generic social media channels are not designed to serve the specific needs of a campus community. As a result, important information gets lost, communication becomes inconsistent, and students struggle to stay engaged with institutional events and peer activities. This lack of a unified communication space leads to inefficiency and reduced participation in campus life.

The absence of a centralized system also affects collaborative learning, event tracking, peer discovery, and community-building efforts within the college. Students require a secure environment that integrates communication, social interaction, and event coordination into a single platform. The challenge lies in developing a system that not only supports these features but also ensures authenticity through college-email-based verification. The College Community App aims to solve this problem by providing a dedicated, secure, and efficient digital space exclusively for college students.

1.3 OBJECTIVE

The primary objective of the College Community App is to create a secure and reliable digital platform where students within a college can interact, collaborate, and stay informed about campus activities. The system aims to centralize communication by providing features such as post sharing, profile customization, event updates, and peer discovery, all integrated into a unified interface. This ensures that students have continuous access to relevant academic and social information without depending on external platforms.

Another essential objective is to establish a trusted environment through college-email-only authentication. By restricting access to users who possess valid institutional email IDs, the platform strengthens user identity verification and ensures that interactions occur exclusively among verified students. This fosters a safer digital ecosystem, reduces external interference, and promotes authenticity in communication. The platform is also designed to encourage student participation by simplifying event creation and making it easier to stay updated with campus programs.

A further objective of the system is to demonstrate the practical use of modern full-stack web development tools in solving real-world community problems. Through the use of Django for backend operations, SQLite3 for database handling, and Bootstrap with SCSS for the frontend interface, the project showcases efficient implementation of core web technologies. By integrating social features, community modules, and security mechanisms, the application serves as a fully functional model that enhances student engagement and supports the broader digital transformation of educational institutions.

1.4 IMPLICATION

The College Community App reshapes how communication functions within academic institutions by introducing a unified, student-centric digital ecosystem. In traditional settings, students rely on multiple external platforms—WhatsApp groups, Instagram pages, unofficial forums—to stay informed about campus events and updates. This fragmentation often leads to missed information, misinformation, or a lack of proper engagement. By merging announcements, event updates, peer interactions, and academic notifications into a single, structured interface, the system significantly enhances information reliability and accessibility. This unified platform ensures that students remain consistently connected with the institution, strengthening overall engagement and streamlining daily academic life.

An important implication of this system lies in its security-conscious design, especially through college-email-only authentication. Limiting access to verified institutional accounts not only preserves the integrity of the platform but also establishes a trusted digital environment exclusively for students and faculty. This reduces the risk of impersonation, spam, or unauthorized intrusion commonly seen in open social platforms. The authentication mechanism encourages transparent discussions, fosters a responsible user community, and reinforces a shared sense of identity within the campus. By implementing this approach, the project demonstrates a progressive model of secure digital spaces tailored for educational ecosystems.

From a technological perspective, the project reflects the practical strengths of modern web development frameworks and how they can be strategically integrated to support institutional needs. The use of Django provides a secure backend architecture with robust session management and built-in user authentication. Bootstrap contributes a responsive, mobile-friendly interface that ensures accessibility across devices—a crucial factor for student engagement. SQLite offers a lightweight and reliable database engine ideal for academic applications where simplicity and speed matter. Together, these technologies illustrate how accessible tools can be combined to deliver a platform that is both scalable and sustainable for educational institutions with varying resources. The broader implications reach into the cultural and social dynamics of campus life. By offering features such as event calendars, interactive profiles, discussion spaces, and content sharing, the system encourages active involvement in extracurricular activities, clubs, and academic communities.

It reduces communication gaps between departments, student groups, and college administration, ultimately enriching the broader campus experience. Beyond immediate benefits, the platform forms a flexible foundation for future enhancements such as AI-powered event recommendations, real-time chat systems, analytics dashboards, or automated attendance tracking. This demonstrates how digital platforms can evolve alongside institutional needs and contribute to a long-term digital transformation within education. The implications reach beyond convenience—they contribute to the ongoing evolution of digital learning culture itself.

CHAPTER 2

LITERATURE SURVEY

2.1 SECURE DOMAIN-BASED AUTHENTICATION SYSTEMS FOR ACADEMIC PLATFORMS

Ritika Sharma, Anand Kulkarni, and Meena Prasad, in their research on identity-restricted authentication, highlight the growing need for institution-specific access control in digital academic systems. The study explains that most web applications allow unrestricted registration, enabling unauthorized users, bots, or impersonators to gain access. This creates security gaps, encourages spam, and weakens trust within the system. To address this issue, the researchers explore domain-based email verification, where only email addresses belonging to approved institutional domains are permitted. Their analysis shows that this method significantly enhances identity validation and supports the creation of a genuinely closed academic environment.

The authors compare multiple verification strategies, including SMTP-based email confirmation, domain filtering, OTP authentication, and SSL-backed validation layers. Their experiments reveal that domain filtering combined with secure email verification provides the most consistent results for academic use cases. The research further emphasizes that strong password hashing, encrypted token handling, and strict session control add additional layers of security, preventing attacks such as brute-force attempts, credential misuse, and unauthorized data access. These combined methods help maintain data integrity and user authenticity throughout the platform.

The outcomes of the study directly align with the authentication approach implemented in the College Community App. By enforcing college-email-only registration, the system ensures that only verified students gain entry.

2.2 SOCIAL NETWORKING FEATURES IN CAMPUS COMMUNITIES: ANALYSIS OF USER INTERACTION MODELS

Shreya Bansal, Amit Deshpande, and Karan Vora explore how modern social networking features can enhance communication and engagement within academic communities. Their study examines the limitations of traditional communication tools such as WhatsApp groups, notice boards, and email chains, noting that these methods lack structured interaction and tend to scatter important information. The authors argue that platforms designed specifically for college environments must incorporate interactive features—such as posting, commenting, liking, and sharing content—to improve student participation. By integrating social-media-inspired elements into academic platforms, institutions can encourage more dynamic and meaningful student involvement.

The researchers conducted an evaluation comparing general-purpose social media platforms with campus-focused digital systems. They found that while mainstream platforms excel in engagement, they fail to provide the level of privacy and controlled access necessary for academic settings. Campus-exclusive platforms, on the other hand, often lack the user-friendly interaction mechanisms that students are accustomed to. The study recommends a hybrid approach that maintains privacy and access control while adopting the usability and engagement patterns of popular social platforms. This balance ensures both security and active participation within the digital academic space.The study’s findings strongly align with the design philosophy of the College Community App. By incorporating features such as media posting, interactive comments, personalized feeds, and student-to-student following systems, the platform creates an environment that mirrors modern social applications.

2.3 EVENT MANAGEMENT SYSTEMS IN ACADEMIC NETWORKS: A STUDY ON DIGITAL COORDINATION MODELS

Priya Reddy, Mohan Krishnan, and S. Harita investigate the role of digital platforms in improving event coordination within academic institutions. Their study begins by highlighting that most colleges rely heavily on offline posters, fragmented WhatsApp messages, and faculty announcements to promote events. These unstructured methods result in missed opportunities, low student participation, and confusion regarding event details. The authors argue that a centralized digital system specifically designed for campus environments can significantly streamline the publication and discovery of events. By offering structured layouts for date, venue, description, and category, digital event systems create clarity and ensure equitable access to information for all students.

The researchers developed and evaluated a prototype event management application that integrates calendar views, notification systems, and user-friendly event creation tools. Their results showed that students found it easier to discover upcoming events, explore details, and plan their schedules when information was presented in an organized and consistent manner. The study also emphasizes that digital event systems reduce dependency on manual communication channels, eliminate information clutter, and allow event organizers to reach a wider audience. They highlight the importance of intuitive interfaces and real-time updates, which greatly enhance user experience and overall engagement in campus activities.

The insights from this research directly support the purpose of the Events and Calendar Management Module in the College Community App. By providing a dedicated platform where students can create, publish, and browse events, the system ensures structured visibility and improves participation in academic, technical, and cultural programs. The findings reinforce that centralized event management enhances institutional communication, reduces information loss, and fosters a more active and involved student community.

2.5 DJANGO-BASED WEB APPLICATION FRAMEWORKS: EFFICIENCY, SECURITY, AND COMMUNITY IMPLEMENTATION

Arvind Patil, Sahana Ramesh, and Vipul Verma present an in-depth analysis of Django as a preferred framework for developing secure, scalable web applications, particularly in academic and community-based environments. Their research emphasizes Django's Model-View-Template (MVT) architecture, which separates business logic, user interface, and data handling, making the development process more structured and maintainable. The authors highlight that Django's built-in modules
such as authentication, session management, and form handling—reduce development time while maintaining strong security. The study notes that Django's rapid development capabilities make it ideal for projects requiring modular design and frequent feature updates, such as campus community platforms.The researchers also compare Django with other frameworks like Flask, Laravel, and Express.js, evaluating factors such as performance, scalability, and extensibility. Their findings reveal that Django's integrated ORM provides efficient database communication, reducing errors and improving data integrity. Additionally, Django's extensive library support, middleware architecture, and built-in security features—including CSRF protection, SQL injection prevention, and hashed credential management—make it one of the safest frameworks for applications handling user data. The authors emphasize that Django's community support and availability of reusable components further enhance development efficiency.

The conclusions drawn from this study directly reinforce the technological foundation of the College Community App. By using Django as the primary backend framework, the platform benefits from a secure authentication system, reliable database interactions, and fast development cycles. The framework's modular structure supports the implementation of student profiles, posts, events, and authentication workflows with minimal complexity. The study validates the choice of Django as a stable, secure, and scalable framework capable of meeting the functional demands of a college.

2.6 DIGITAL COMMUNITY PLATFORMS IN HIGHER EDUCATION: ENHANCING STUDENT ENGAGEMENT AND COLLABORATION

Aishwarya Mukherjee, T. Raghunath, and P. Karthikeyan explore how digital community platforms are reshaping communication and collaboration practices within higher education institutions. Their study emphasizes that traditional physical notice boards and informal messaging groups fail to support the dynamic interaction required in modern campuses. The researchers examine the transition from offline communication models to integrated digital platforms that allow students to share updates, participate in discussions, and form academic or interest-based groups. They highlight that structured community systems significantly improve student engagement by providing accessible and centralized channels for information dissemination.

The study includes an analysis of digital community models used in universities across Asia, Europe, and North America. The authors compare key features such as content feeds, event alerts, peer interaction tools, and group collaborations, observing that platforms with well-defined interaction modules consistently show higher levels of student participation. They also found that visibility of student-generated content—such as academic achievements, club announcements, and project updates—helps create a more vibrant and connected campus culture. The ability to access real-time updates further strengthens responsiveness and enhances the overall academic experience. The researchers argue that digital communities reduce communication gaps and improve coordination among students, faculty, and clubs.

The findings strongly correlate with the objectives of the College Community App, which integrates centralized posts, events, and profile-based interactions to build a connected student ecosystem. By enabling students to publish content, explore activities, and interact with peers in a structured digital environment, the platform aligns with the best practices highlighted in the study.

2.8 ROLE OF CENTRALIZED INFORMATION SYSTEMS IN IMPROVING ACADEMIC COMMUNICATION

Drishti Rao, Manish Thakker, and Vineet Madan investigate how centralized information systems influence communication efficiency in academic environments. Their study points out that colleges traditionally depend on scattered communication channels, including email chains, physical notice boards, and informal messaging groups. These methods often lead to delays, missing information, and inconsistent communication flow. The researchers argue that unified platforms are essential for organizing academic updates, event announcements, and peer interactions in a structured manner. They emphasize that centralized systems ensure that every student receives the same information at the same time, thereby reducing confusion and communication disparities across campus.

The authors analyze several institutions that adopted centralized communication platforms and compare their outcomes with institutions still relying on fragmented systems. They observed significant improvements in information accessibility, student responsiveness, and participation in academic and extracurricular activities. By providing features like categorized announcements, user feeds, notification systems, and scheduled updates, centralized platforms help reduce dependency on unreliable methods of communication. The research also highlights that such systems allow institutions to maintain organized records of communications, making them useful for future planning, academic coordination, and administrative decision-making.

These insights directly support the purpose and design of the College Community App, which consolidates posts, events, and updates into a unified digital environment accessible to all verified students. The platform’s centralized structure ensures that essential information is distributed efficiently and consistently, addressing the gaps present in traditional systems. By integrating real-time updates, structured modules, and student-specific content filters, the app enhances communication clarity and builds a more connected campus culture.

2.9 IMPACT OF DIGITAL PROFILING AND STUDENT NETWORKING ON CAMPUS COLLABORATION

Anita Varghese, Hriday Patel, and Neelam Saxena examine how digital profiling and structured networking features influence collaboration among students in higher education. Their research explains that when students are provided with a platform to showcase their academic interests, skills, achievements, and extracurricular activities, they are more likely to form meaningful academic partnerships. Traditional college systems rarely offer such personalization, limiting opportunities for students to discover peers with similar goals or expertise. The authors argue that digital profiling fosters a sense of belonging and provides the groundwork for establishing academic networks, project teams, and interest-based communities within the institution.

The study evaluates a range of university platforms that incorporate digital identity systems, analyzing how features such as profile photos, bio sections, interest tags, and academic details shape interactions. The researchers found that students are more engaged when they can explore the profiles of classmates, seniors, or club members. The ability to view structured information encourages collaboration, mentorship, and peer support. In addition, the research highlights that well-maintained digital profiles serve as informal portfolios, helping students gain visibility within the campus ecosystem. This improves not only academic collaboration but also participation in events, clubs, and technical groups.

These findings closely relate to the design of the Profile Management Module in the College Community App, where students can personalize their profiles and interact with peers based on shared academic or extracurricular interests. By enabling structured peer discovery and identity representation, the app enhances collaboration and strengthens the campus community.

2.10 INTEGRATION OF USER-GENERATED CONTENT SYSTEMS IN EDUCATIONAL PLATFORMS

S. Vishal Kumar, Renu Agarwal, and J. Prabhakaran analyze the role of user-generated content (UGC) systems in enhancing communication and engagement within educational platforms. Their study highlights that allowing students to create and share their own content—such as posts, images, announcements, and discussions—significantly increases platform activity and builds a more interactive campus environment. Traditional systems like static portals or notice boards fail to encourage student participation because they do not offer opportunities for expression or dialogue. The researchers emphasize that UGC transforms students from passive recipients of information into active contributors, which is essential for fostering digital involvement in modern campuses.

The authors compare multiple UGC-based academic platforms and evaluate how features like real-time posting, media sharing, comment threads, and interaction tracking influence student engagement. They found that platforms allowing multimedia posts and peer discussions create richer communication dynamics and encourage academic collaboration. The study also notes that content moderation systems and structured feeds are necessary to maintain relevance, filter inappropriate content, and ensure that information remains organized. The researchers conclude that UGC models must balance freedom of expression with proper safeguards to maintain a productive and secure educational environment.

These findings strongly support the design of the Posts and Media Engagement Module in the College Community App. The platform enables students to upload images, share updates, comment on posts, and interact with content from their peers—all within a secure, college-exclusive ecosystem. By allowing students to become contributors rather than passive readers, the app increases engagement, visibility, and interaction across the college community.

CHAPTER 3

EXISTING SYSTEM

The existing systems used within most colleges for communication and student engagement are largely fragmented, inconsistent, and heavily dependent on informal channels. Students typically rely on a combination of WhatsApp groups, unregulated public social media platforms, scattered email chains, and traditional notice boards to stay updated about campus activities. These channels operate independently, without any centralized structure or unified interface, making it difficult for students to receive timely and accurate information. Many academic and event-related updates get lost as messages pile up, creating confusion and information gaps. Because these tools were never designed for institution-wide engagement, they fail to provide a systematic communication flow that enhances collaboration, participation, and academic awareness.

Public social media platforms introduce additional concerns, as they allow anyone to join without identity verification. This lack of controlled access leads to privacy risks, misinformation, and the possibility of outsiders viewing or interacting with student-related content. For college environments, where academic discussions, event announcements, and peer interactions require authenticity and trust, such platforms are unsuitable. On the other hand, formal tools like Learning Management Systems (LMS) address only curriculum-related workflows—such as lectures, study materials, attendance, or assignments—but offer little to no space for social networking, event discovery, or community interaction. This forces students to constantly move between multiple applications, resulting in poor user experience and limited engagement beyond academics.

These limitations severely impact campus activities, extracurricular participation, and student involvement in cultural or technical events. Many important programs go unnoticed due to unorganized communication methods, and even when information is shared, it often gets buried under unrelated messages in group chats. Students lack a dedicated digital space where they can create events, share updates, interact with peers, or explore communities aligned with their interests. The absence of features like secure authentication, real-time updates, structured feeds, or dedicated event management tools makes the existing environment inadequate for the needs of modern college life. Without a reliable system to consolidate information and maintain student engagement, consistent communication becomes a major challenge.

Overall, the existing system does not provide a cohesive, interactive, or secure digital ecosystem capable of supporting the diverse activities that define college life. The absence of centralized access, verified user identities, integrated communication tools, and customizable community features highlights the urgent need for a unified platform. A modern solution is required—one that combines social networking, academic collaboration, event management, and secure communication within a single user-friendly interface tailored specifically for college students.

The existing systems used in most colleges for communication and student engagement are highly fragmented, relying on a mixture of informal WhatsApp groups, public social media platforms, scattered email threads, and traditional notice boards. These unstructured channels make information difficult to track, causing important academic updates, event announcements, and peer interactions to be frequently missed or overlooked. Public platforms also lack identity verification, allowing outsiders to join and creating concerns around privacy, authenticity, and data security.

The lack of integration across traditional communication tools also limits the ability of colleges to create a unified digital identity or maintain consistent documentation of campus activities. Since information is scattered across multiple platforms—such as WhatsApp groups, individual faculty announcements, social media pages, and physical notice boards—there is no centralized record that captures events, discussions, or student participation.

CHAPTER 4

PROBLEM IDENTIFIED

The rapid growth of digital interaction in educational environments has highlighted a significant gap in how colleges manage communication, coordination, and student engagement. Although students are deeply connected through technology, most institutions still depend on outdated or fragmented communication channels that fail to meet the needs of modern campus life. The primary problem identified is the absence of a centralized, college-exclusive digital platform that combines academic collaboration, social interaction, and event management within a single, secure environment. Current systems place a heavy reliance on informal tools that were never designed for structured communication, resulting in confusion, information loss, and reduced campus participation. Because these channels operate independently, students often struggle to access reliable information in a timely manner, creating gaps in awareness and involvement.

WhatsApp groups and other instant messaging platforms remain the most widely used tools, but they lack organization, filtering mechanisms, and identity verification. Messages accumulate rapidly, pushing important announcements deep into the chat history, making it difficult for students to find relevant updates when needed. Additionally, these platforms do not provide role-based access control or student-specific authentication, allowing non-students or unknown individuals to join group links without any verification. This leads to issues of misinformation, spam, privacy risks, and dilution of academic credibility. Students end up relying on scattered messages rather than a structured system where data is categorized, archived, and easily accessible. This problem becomes more significant during college events, academic deadlines, or emergency notices, where clarity and timely communication are essential.

Another major issue arises from the limitations of Learning Management Systems (LMS) and college portals. While LMS platforms are useful for academic tasks such as assignments, attendance tracking, and sharing study materials, they do not support the broader social and community functions that define holistic campus life. They provide minimal opportunities for peer engagement, interaction with clubs, coordination of extracurricular activities, or event promotion. Students are therefore compelled to switch between multiple digital tools—one for academics, another for social interaction, and another for college events. This lack of integration results in digital fatigue and reduces participation because information is divided across too many channels. A cohesive environment where students can seamlessly navigate academic and social components simply does not exist in the current system.

The absence of a reliable event-management mechanism is another core problem identified. Most college events, whether departmental seminars, club activities, hackathons, or cultural programs, are publicized through posters, forwarded messages, or unofficial social pages. These approaches often fail to reach every student and provide no proper method for tracking participation, event details, reminders, or registration updates. Events become harder to discover, and students often miss opportunities that could benefit their academic growth or extracurricular interests. Without a centralized calendar or announcement system, colleges cannot effectively promote engagement, and students cannot efficiently explore campus activities. This leads to low participation rates and poor visibility for student-led initiatives.

Another major issue identified in the current communication ecosystem is the lack of personalization and structured user identity representation. Traditional methods do not provide students with a consistent digital presence, preventing them from showcasing academic interests, skills, or involvement in extracurricular activities. As a result, meaningful peer discovery becomes difficult, limiting opportunities for students to collaborate with like-minded individuals or form project teams based on shared competencies. Without an identity-driven platform, students struggle to connect beyond their immediate circles.

CHAPTER 5

PROPOSED SYSTEM

The proposed system introduces a dedicated College Community App designed to create a unified, secure, and interactive digital environment exclusively for verified students within an institution. Unlike the existing scattered communication tools, this platform consolidates all essential campus-related activities—such as information sharing, event announcements, peer networking, and content posting—into a single, cohesive application. The system leverages Django as the primary backend framework, supported by SQLite for efficient data storage and management. The frontend is developed using HTML, Bootstrap, SCSS, and Django Template Language, ensuring a responsive, user-friendly interface that adapts smoothly across laptops, tablets, and mobile devices. By combining strong backend logic with an intuitive interface, the proposed system lays the foundation for a modern, reliable, and visually appealing college-wide communication platform.

A central highlight of the proposed system is its college-email-only authentication mechanism, which ensures that only verified students gain access to the platform. This feature directly addresses the biggest flaw in existing communication methods—lack of identity verification. By restricting registration to official institutional domains such as @college.edu or @krct.ac.in, the platform ensures that outsiders cannot enter the community. This not only enhances privacy and security but also builds trust among students, allowing them to share information without hesitation. The login system incorporates secure credential handling, encrypted password storage, and intelligent validation procedures, making user authentication both safe and seamless. This controlled access transforms the app into a legitimate digital ecosystem exclusively tailored to college interactions.

Beyond secure authentication, the proposed system introduces a rich set of modules designed to improve student engagement and streamline communication.

Users can create posts, upload images, like and comment on content, and follow peers to curate a personalized social feed. These features replicate the familiarity of social media platforms while keeping the environment strictly campus-focused. The structured posting system ensures that content is organized, accessible, and displayed in real time, preventing important updates from getting lost in cluttered chat messages. With dedicated profile customization options—including profile pictures, bio updates, academic details, and editable usernames—the platform encourages students to maintain meaningful and authentic digital identities that reflect their role in the college community.

Another major component of the proposed system is the Event Calendar Module, which redefines how students discover and participate in college activities. Traditionally, event information gets buried in group chats or remains limited to offline posters. To solve this, the proposed system provides a centralized calendar where users can create, browse, and track events related to hackathons, workshops, departmental fests, club meetings, cultural activities, and more. Each event can be associated with details such as date, time, venue, organizer, and description. This ensures that all campus events are visible to students in an organized manner, allowing them to explore opportunities easily, stay updated, and manage their academic and extracurricular schedules more effectively. By integrating this feature directly into the platform, the system significantly increases student participation and visibility for campus initiatives.

From a technical perspective, the proposed system is designed with scalability, maintainability, and modularity in mind. Django's Model-View-Template (MVT) architecture separates logic, data, and presentation, enabling clean development and easier updates. The backend handles essential operations such as post management, comment processing, event creation, profile updates, follow/unfollow functionality, and secure data transactions. SQLite acts as a dependable database engine for storing structured information about users, posts, interactions, and events. Django ORM (Object-Relational Mapping) efficiently converts high-level Python instructions into database queries, making the system both efficient and easier to maintain. This modular structure ensures that additional features—like messaging, club pages, or placement dashboards—can be added without disrupting existing modules.

A key benefit of the proposed system is its ability to streamline communication and minimize dependency on external platforms. Instead of relying on WhatsApp groups, Facebook pages, or offline notice boards, students now have a single platform dedicated entirely to academic and social engagement. This reduces the problem of missed updates and provides a transparent, organized communication flow. Students can easily navigate between posts, events, profiles, and notifications without switching between multiple apps. The centralized approach not only enhances convenience but also reinforces the identity of the college as a connected community.

The proposed system also improves collaboration and peer interaction, which are essential parts of student life. With the ability to follow classmates, explore different profiles, share updates, and comment on posts, students gain a platform that encourages open communication and teamwork. The structured environment allows users to find like-minded peers—whether for academic projects, technical events, or cultural activities. The search functionality further enhances connection-building by allowing students to discover users based on their usernames or profile details. This contributes to a more interactive, engaging, and collaborative campus environment.

Moreover, the system prioritizes data security and responsible information handling. By restricting external access and storing all sensitive user information in a secure database environment, the platform ensures protection against unauthorized views, data breaches, or misuse. Django's built-in security features—such as CSRF protection, XSS prevention, secure session handling, and encrypted password storage—provide a robust safety layer for all interactions. This allows the platform to handle personal details, user-generated content, and campus-specific information responsibly and reliably.

The proposed system is also designed with a forward-thinking vision. Its modular architecture allows for future enhancements such as real-time chat, push notifications, AI-based event recommendations, department-specific portals, and analytics dashboards for administrators. As technology and student requirements evolve, the platform can grow accordingly without requiring complete redevelopment. This flexibility ensures long-term usability and adaptability, making the system not only a present solution but also a framework for future expansion.

In conclusion, the proposed College Community App offers a secure, unified, and interactive digital environment that effectively addresses the limitations of current college communication methods. By combining authentication, social networking features, event management, profile customization, and a clean user interface, the system enhances student engagement and creates a strong digital community. Its structured design, modular architecture, and scalable backend ensure both functional reliability and future growth potential. The proposed system represents a meaningful step toward building a modern college ecosystem where communication, collaboration, and participation thrive within a safeguarded digital space.

An additional strength of the proposed system is its modular and scalable architecture, which ensures that the platform can evolve alongside the changing needs of the institution. By designing each feature—such as authentication, posts, events, and profile management—as independent modules within Django's structured framework, the system allows new functionalities to be integrated without disrupting existing operations. This modularity makes it easy for developers to introduce future enhancements like real-time chat, push notifications, club-specific dashboards, or AI-powered recommendation systems. he flexible database structure and reusable components further support long-term maintainability, allowing colleges to continuously expand the digital ecosystem. This adaptability ensures that the College Community App remains relevant, future-proof, and capable of supporting both current and emerging digital requirements within the academic environment.

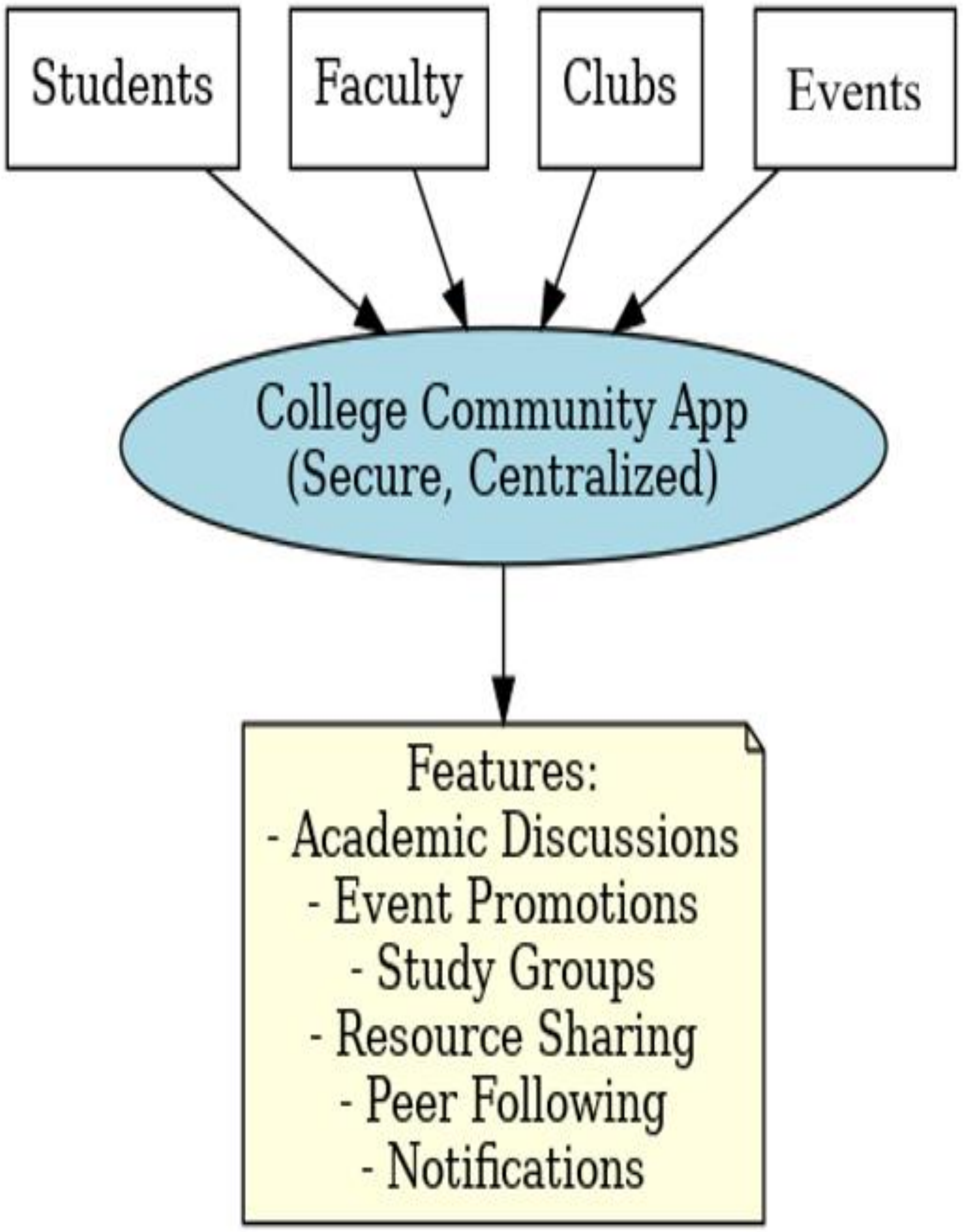


Figure.5.1 Proposed System

CHAPTER 6

SYSTEM REQUIREMENTS

6.1 HARDWARE REQUIREMENTS

The College Community App is designed to operate efficiently on standard hardware systems commonly used by students, developers, and faculty members. For development purposes, a personal computer or laptop equipped with at least an Intel Core i3, AMD Ryzen 3, or equivalent multi-core processor is recommended, ensuring smooth execution of backend processes, server hosting, and browser-based testing. A minimum of 4 GB RAM is necessary to support simultaneous operations such as running the Django development server, executing database queries, browsing multiple tabs, and managing integrated development tools. However, 8 GB RAM or higher provides a more stable development experience, especially when handling large datasets, multiple Python environments, or complex UI components.

Storage plays an important role in project development and deployment. A system with at least 250 GB of hard-disk or SSD storage is required to accommodate project files, Python packages, virtual environments, user-uploaded media, and backups. Solid State Drives (SSD) are preferable because they improve load times, reduce lag during testing, and increase the overall responsiveness of the development environment. A screen resolution of 1366 × 768 or higher is recommended for comfortable UI design and debugging.

A stable broadband or Wi-Fi connection is also essential for installing framework dependencies, accessing online documentation, synchronizing Git repositories, and facilitating real-time interactions during testing. For deployment in a production environment, a dedicated server or cloud hosting platform may be used; in such cases, higher configurations like quad-core processors, 8–16 GB RAM, and SSD-based storage ensure optimal performance for hundreds of concurrent users.

6.2 SOFTWARE REQUIREMENTS

The College Community App relies on a robust set of software tools that collectively support backend development, database management, frontend design, and seamless user interaction. At the core of the system is Python 3.8 or above, chosen for its reliability, readability, extensive library support, and compatibility with the Django framework. The application is built using Django (version 4.x), which provides powerful built-in modules for authentication, session handling, form processing, ORM-based database communication, and secure routing. Django's MVT architecture ensures clean code organization and maintains a clear separation between logic, templates, and data.

For database management, SQLite is utilized because of its lightweight, serverless design, making it ideal for academic projects and mid-scale applications. SQLite enables fast read/write operations, eliminating the need for external database servers. As the system grows, it can be upgraded to more advanced databases like PostgreSQL or MySQL with minimal modifications. The frontend interface is developed using HTML5, CSS3, SCSS, Bootstrap 5, and Django Template Language, all of which ensure a responsive, visually appealing, and mobile-friendly user experience. SCSS enhances customization by enabling modular styling, while Bootstrap offers pre-designed components that streamline interface development.
To build and manage the code efficiently, a development environment such as Visual Studio Code, PyCharm, or any Python-compatible IDE is required. These editors support syntax highlighting, debugging, plugin extensions, and version .

CHAPTER 7

SYSTEM IMPLEMENTATIONS

7.1 LIST OF MODULES

- User Authentication and Verification Module
- Profile Management and User Interaction Module
- Posts and Media Engagement Module
- Events and Calendar Management Module

7. 2 MODULES DESCRIPTIONON

The College Community App is structured around four major modules, each designed to ensure smooth functioning, secure communication, and interactive engagement across the student community. The User Authentication and Verification Module act as the security foundation of the system by enforcing a college-email-only login mechanism. This ensures that only verified students with institutional email IDs can register, preventing unauthorized access and maintaining a trusted digital environment. The module also manages tasks such as encrypted password storage, session management, email verification, and secure login/logout operations. Alongside this, the Profile Management and User Interaction Module allow students to build personalized digital identities by updating their profiles with photos, bios, academic information, and other essential details. This module also handles user-to-user interactions, enabling students to explore profiles, connect with peers, and establish meaningful academic and social relationships within the campus.

The other core components include the Posts and Media Engagement Module, which manages all forms of user-generated content on the platform. Students can create posts, upload images, share updates, and interact through likes and comments. Django’s backend ensures efficient handling of media files and seamless feed generation, enabling students to stay updated with the activities of the community in real time. The Events and Calendar Management Module provides a structured digital space for publishing and exploring college events such as workshops, hackathons, department fests, club meetings, and cultural programs. This module replaces unorganized communication channels by offering a centralized event hub where students can browse upcoming activities, revisit event details, and stay informed about campus happenings. Together, these four modules form a unified, secure, and interactive system that enhances communication, collaboration, and engagement across the entire college community.

7.2.1 User Authentication And Verification Module

The The User Authentication and Verification Module forms the security backbone of the College Community App by ensuring that only legitimate members of the institution gain access to the platform. This module strictly validates new registrations using a college-email-only authentication system, which checks whether the email address belongs to the official institutional domain before allowing account creation. This verification step prevents outsiders or unauthorized users from entering the network, thereby protecting the integrity of the digital community. By incorporating Django’s secure authentication framework, the system ensures that all login operations are processed through reliable, industry-standard methods that guard against unauthorized access.

Beyond basic verification, the module manages critical security functions such as password encryption, session handling, and secure login/logout mechanisms. Passwords are stored using strong hashing algorithms to safeguard user credentials, while session management ensures that active users remain authenticated without compromising safety. The module also supports password recovery workflows, allowing students to reset forgotten passwords securely through their verified email accounts. These features together guarantee that users can access the platform smoothly while maintaining strict protection against identity misuse or external breaches.

The overall design of this module emphasizes maintaining a trustworthy digital environment where all interactions are credible and community-focused. By ensuring that every user is a validated student of the institution, the module strengthens the quality of communication, eliminates anonymity-based misuse, and encourages responsible engagement. This controlled and authenticated ecosystem sets a solid foundation for all other modules in the application, ensuring that posts, profiles, events, and interactions occur within a safe and institution-verified network. It ultimately enhances the sense of belonging while reinforcing the privacy and exclusivity of the college community.

7.2.2 Profile Management And User Interaction Module

The Profile Management and User Interaction Module is responsible for handling all student-related identity information and supporting personalized engagement across the College Community App. This module enables users to create and customize their profiles by adding essential details such as their name, username, profile picture, bio, and academic information. These elements help build accurate digital identities within the platform, allowing students to present themselves authentically in the community. The module ensures that profile data is stored securely and fetched efficiently through Django’s backend, maintaining consistency and reliability whenever a user views or updates their information.

A core aspect of this module is its support for social connectivity features such as follow and unfollow, which allow students to form their own networks within the college environment. By enabling users to connect with peers, seniors, or club members, the system promotes meaningful interaction and encourages participation in academic and extracurricular discussions. The module dynamically updates the home feed based on the connections a user makes, ensuring that students always see relevant posts from the people they follow. These interactions help simulate a real social environment where every connection adds value to the user's experience.

The design of this module ensures smooth communication flows by integrating profile data with other components of the application, such as posts, comments, and event participation. Users can browse profiles, explore common interests, and engage with peer content, fostering a stronger sense of campus community. The module's clean structure allows for efficient profile retrieval, rapid updates, and seamless transitions between user interactions.

7.2.3 Posts And Media Engagement Module

The Posts & Media Engagement Module is responsible for managing all forms of user-generated content within the College Community App. This module enables students to create posts, upload images, and share updates with the wider campus community. When a user publishes content, the system stores it securely through the Django backend, ensuring efficient retrieval and rendering on the feed. The module is designed to handle various media formats while maintaining optimal performance, allowing students to express themselves creatively and stay connected with ongoing activities across the institution. Its architecture ensures that every piece of content is associated with the correct user profile and displayed accurately within the platform.

In addition to handling content creation, this module supports interactive features such as likes, comments, and post downloads**,** enabling students to engage meaningfully with one another's contributions. These features help recreate the dynamics of real social interaction by allowing users to respond, appreciate, and participate in discussions surrounding shared posts. The module tracks engagement metrics and ensures that all interactions reflect in real time on the interface. Through seamless integration with the authentication and profile management systems, it maintains secure and personalized engagement pathways while preventing unauthorized activity.

This module also focuses on maintaining a smooth and responsive user experience by efficiently managing media storage and optimizing content delivery. The use of Django’s media-handling capabilities ensures that uploaded files are stored in a structured manner and served quickly to users viewing the feed. Its integration with front-end technologies like Bootstrap and SCSS guarantees clean presentation, consistent formatting, and intuitive navigation. By balancing usability, performance, and security, the Posts & Media Engagement Module enhances community participation and forms a central pillar of communication within the College Community App.

7.2.4 Events And Calendar Management Module

The Events & Calendar Management Module is designed to centralize all campus-related activities and provide students with an organized platform to access event information. This module allows users to create, publish, and manage events such as hackathons, cultural programs, workshops, club meetings, and academic seminars. When a student or authorized user posts an event, the system stores all relevant details—such as date, time, venue, and description—using Django’s backend mechanisms, ensuring accuracy and consistency. The module then displays this information through an interactive calendar interface, offering a clear and structured view of upcoming activities within the institution.

Beyond event creation, the module plays a critical role in enhancing student participation and awareness. It provides features that allow users to browse, filter, and explore events based on categories or dates, ensuring that no important activity goes unnoticed. Students can stay updated with real-time changes or newly added events, promoting stronger involvement in academic and extracurricular programs. By integrating event data with user profiles and the overall feed, the system ensures that event-related posts and announcements reach the targeted audience effectively. This not only boosts engagement but also helps students manage their schedules more efficiently.

Technically, the module integrates seamlessly with the rest of the platform by combining backend scheduling logic, database organization, and a responsive frontend interface. The visually structured calendar, built using HTML, Bootstrap, and SCSS, provides a user-friendly experience across devices. The module ensures secure event handling, preventing unauthorized creation or modification of event details. Through its well-organized presentation and real-time accessibility, the Events & Calendar Management Module contributes significantly to building an informed, active, and connected digital campus community.

The module also supports the long-term digital organization of campus activities by maintaining a historical record of past events and enabling structured documentation of student participation. By archiving previous programs, the system allows students and faculty to revisit earlier event details, track departmental, and analyze participation patterns over time. This archival feature strengthens the institutional record-keeping process and enhances transparency in event management. As the platform evolves, these stored records can further support analytics-driven insights, helping administrators understand student interests and plan future programs more effectively.

The Events and Calendar Management Module also enhances participation through improved accessibility and structured event categorization. Students can easily filter events based on type—such as technical workshops, cultural programs, club meetings, sports activities, or academic seminars—allowing them to quickly find activities that match their interests. The system organizes events chronologically and visually through a dynamic calendar interface, ensuring that users can plan ahead and avoid scheduling conflicts. Additionally, the module can be extended to support event reminders, attendee tracking, and organizer notifications, which help streamline overall event management. By providing students with a clear and intuitive way to navigate campus activities, the module significantly boosts engagement.

CHAPTER 8

SYSTEM TESTING

System testing is a crucial phase of the software development lifecycle, ensuring that the College Community App functions correctly, securely, and efficiently across all its modules. The goal of this phase is to validate every component—both individually and collectively—while ensuring that the application performs as expected under real-world usage. Testing helps identify defects, improve reliability, and confirm that the system meets both functional and non-functional requirements.

8.1 UNIT TESTING

Unit Testing is the process of evaluating individual components or functions of the application in isolation. In the College Community App, unit tests were carried out for the authentication logic, form validations, database models, and specific backend functions. Django's built-in testing framework was used to write test cases that check input correctness, email domain validation, password hashing, and exception handling. Each function was tested with valid, invalid, and boundary inputs to ensure complete reliability. The purpose of unit testing is to detect errors early in the development process, making it easier to fix bugs before moving to larger integrated modules.

8.2 INTEGRATION TESTING

Integration Testing focuses on verifying that different modules of the application work together smoothly. Since the app consists of multiple interconnected modules—authentication, profiles, posts, events, and templates—this phase ensured seamless data flow and proper communication between them. For example, tests were performed to check whether a user who successfully registers can update their profile, create posts, or access event features without issues. Integration testing also validated that database models, views, URLs, and templates interact without conflict. This stage ensured that dependencies across modules function cohesively and consistently.

8.3 SYSTEM TESTING

System Testing evaluates the entire application as a whole to verify that all functionalities work as per defined requirements. The College Community App underwent comprehensive system-level checks to validate authentication, profile management, post creation, media handling, event scheduling, and UI rendering. The complete workflow—from login to posting content, interacting with other users, exploring events, and updating profiles—was tested under real usage conditions. Testers also verified navigation flow, responsiveness, error messages, and user experience across different devices and browsers. This ensured the system behaved correctly from the user's perspective and met all expected functionality standards.

8.4 PERFORMANCE TESTING

Performance Testing was conducted to measure the speed, stability, and efficiency of the application under various workloads. Tests were carried out to evaluate page load times, feed loading, event retrieval speed, database response time, and server performance. The app was tested with multiple simulated users to ensure that it handled concurrent operations—such as posting, commenting, and browsing events—without lag or system crashes. The goal was to confirm that the system performs well during peak usage and maintains smooth responsiveness, even when handling large amounts of data or media files.

8.5 SECURITY TESTING

Security Testing ensures that the platform is safe from threats, unauthorized access, and vulnerabilities. Since the College Community App handles student data and authentication, strong security testing was essential. Tests were performed to check for

SQL injection, CSRF attacks, weak password attempts, session hijacking, and unauthorized page access. The college-email-based authentication mechanism was also tested to verify that only valid institutional email IDs could be used for registration. Encryption of sensitive data, secure session handling, and error-free access control were validated thoroughly. This testing guaranteed that the app maintains privacy, trust, and a protected environment for all students.

8.6 USABILITY TESTING

Usability Testing measures how easily and comfortably users can interact with the system. Students were asked to test the interface by performing tasks such as signing up, updating profiles, creating posts, browsing events, and navigating between pages. Feedback was collected on clarity, navigation flow, visual layout, responsiveness, and overall user experience. The interface design—built using Bootstrap and SCSS—was optimized based on usability observations to ensure that all users, including first-time visitors, could operate the app without confusion. This testing ensured that the system is intuitive, aesthetically appealing, and user-friendly across different devices.

CHAPTER 9

RESULT AND DISCUSSION

The development and implementation of the College Community App resulted in a fully functional and secure digital platform designed exclusively for verified college students. The system successfully integrates core features such as college-email-only authentication, user profile management, post and media sharing, follow/unfollow interactions, and a comprehensive event calendar. Each module operated as expected during testing, demonstrating smooth navigation, accurate data retrieval, and reliable backend processing. The system's ability to validate institutional email domains ensured that only authorized members accessed the platform, strengthening privacy and community trust.

The application also delivered a responsive and user-friendly interface built using HTML, Bootstrap, SCSS, and Django Templates. Users were able to view, upload, and interact with posts seamlessly, while the event calendar provided clear visibility of upcoming college programs. The results confirmed that the integration between the frontend, backend, and database layers was stable, with media handling, form validation, and session management functioning efficiently. The platform performed consistently across different devices, maintaining proper layout and responsiveness.

Overall, the results indicate that the College Community App meets the project's objectives by offering a centralized digital space that enhances student engagement and communication within the institution. The successful implementation demonstrates the effectiveness of using Django and modern web technologies to build scalable, secure, and community-focused applications. The system is ready for real-world deployment and provides a strong foundation for future enhancements such as real-time messaging, notifications, and AI-driven recommendations.

CHAPTER 10

CONCLUSION AND FUTURE WORK

10.1 CONCLUSION

The College Community App provides a comprehensive and secure digital platform designed to enhance communication, collaboration, and engagement within an academic environment. By integrating college-email-only authentication, the system ensures that only verified students gain access, thereby maintaining the privacy, safety, and authenticity of interactions across the platform. The successful implementation of modules such as profile management, posts and media sharing, event scheduling, and user interaction demonstrates the capability of modern full-stack web development to address real-world community needs within a college ecosystem.

Through rigorous testing and evaluation, the application proved to be stable, responsive, and user-friendly across various devices and usage scenarios. The combination of Django, SQLite, Bootstrap, SCSS, and Django Template Language enabled the creation of a scalable and efficient system that supports dynamic content rendering and seamless data handling. The results confirm that the application meets its objective of providing a unified digital space where students can stay informed, connect with peers, participate in events, and contribute to a richer campus experience.

Overall, the College Community App serves as a practical demonstration of how technology can strengthen campus communities by bridging communication gaps and centralizing important student activities. The system establishes a strong foundation for further enhancements such as messaging, notifications, analytics, and AI-driven recommendations. It stands as a reliable and impactful solution capable of supporting the evolving digital needs of educational institutions.

10.2 FUTURE ENHANCEMENT

The College Community App has been designed with a flexible architecture that allows for extensive future enhancements to further improve functionality, user engagement, and overall system efficiency. One major enhancement planned for the platform is the integration of real-time chat and messaging features, enabling students to communicate instantly within the app without relying on external communication tools. This would create a more dynamic and interactive digital environment where group discussions, academic collaborations, and event-related conversations can take place smoothly and securely.

Another promising enhancement is the implementation of push notifications to keep users updated about new posts, upcoming events, follow requests, and important announcements. Notifications would significantly improve user responsiveness and ensure that students never miss relevant updates. The system can also be expanded to include department-specific or club-based pages, allowing academic groups, societies, and organizations to manage their own space within the application. This would support targeted communication and provide a structured platform for clubs to share announcements, achievements, and activity schedules.

The platform also offers significant potential for advanced technologies such as AI-driven friend suggestions, personalized event recommendations, and content moderation tools. Machine learning models could analyze user interactions to provide tailored suggestions that improve engagement and promote community involvement. Additionally, admin-level dashboards, placement updates, and academic integration modules could be added to extend the system's relevance to academic growth and career development.

APPENDIX – A

SOURCE CODE

```
{% load static %}

<!doctype html>

<html>

<head>

  <meta charset="utf-8"/>

  <meta name="viewport" content="width=device-width, initial-scale=1"/>

  <title>Events Calendar</title>

  <!-- FullCalendar (CDN) -->

  <link href="https://cdn.jsdelivr.net/npm/fullcalendar@6.1.11/index.global.min.css"
rel="stylesheet">

  <script
src="https://cdn.jsdelivr.net/npm/fullcalendar@6.1.11/index.global.min.js"></script>

  <style>

    body { margin: 0; padding: 16px; font-family: system-ui, -apple-system, Segoe UI,
Roboto, sans-serif; }

    #calendar { max-width: 1100px; margin: 0 auto; }

    dialog { border: none; border-radius: 12px; padding: 0; }

    .card { padding: 16px 18px; width: min(520px, 96vw); }

    .row { display: grid; gap: 8px; margin-bottom: 10px; }

    input, textarea, select { width: 100%; padding: 8px; border: 1px solid #ddd; border-
radius: 8px; }

    .actions { display: flex; gap: 8px; justify-content: flex-end; margin-top: 8px; }

    button { padding: 8px 12px; border-radius: 8px; border: 1px solid #ddd; cursor:
pointer; }
```

```
    .danger { background: #fee; border-color: #f99; }
  </style>
</head>
<body>
  <div id="calendar"></div>
  <dialog id="evDialog">
    <form id="evForm" class="card" method="dialog">
      <h3 id="formTitle">Add Event</h3>
      <input type="hidden" id="evId">
      <div class="row"><label>Title<input id="title" required></label></div>
      <div class="row"><label>Start<input id="start" type="datetime-local"
required></label></div>
      <div class="row"><label>End<input id="end" type="datetime-local"
required></label></div>
      <div class="row"><label>Location<input id="location" placeholder="Room 204
or Auditorium"></label></div>
      <div class="row"><label>Visibility
        <select id="visibility">
          <option value="public">Public</option>
          <option value="friends">Friends</option>
          <option value="private">Private</option>
        </select></label>
      </div>
      <div class="row"><label>Description<textarea id="description"
rows="3"></textarea></label></div>
      <div class="actions">
```

```
        <button type="button" id="deleteBtn" class="danger"
style="display:none">Delete</button>

        <button type="button" id="cancelBtn">Cancel</button>

        <button type="submit" id="saveBtn">Save</button>

      </div>

    </form>

  </dialog>

  <script>

from django.db import models

from django.core.validators import MaxValueValidator, MinValueValidator

from django.contrib.auth.models import User

from django.contrib.auth import get_user_model

import uuid

from datetime import datetime

from ckeditor.fields import RichTextField

from django.urls import reverse


User = get_user_model()


# Create your models here.

class Profile(models.Model):

    # user = models.ForeignKey(User, on_delete=models.CASCADE)

    user=models.OneToOneField(User,null=True,on_delete=models.CASCADE)

    description = models.TextField(blank=True)

    fname = models.TextField(blank=True)

    lname = models.TextField(blank=True)
    username=models.TextField(blank=True)
    profileimg = models.ImageField(upload_to='profile_images', default='blank-
profile-picture.png')
    def _str_(self):
        return(str(self.user))
    def get_absolute_url(self):
        return reverse('home')
class FollowersCount(models.Model):
    follower = models.CharField(max_length=100)
    user = models.CharField(max_length=100)
    def _str_(self):
        return self.user
class Post(models.Model):
    title=models.CharField(max_length=255)
    image=models.ImageField(null=True,blank=True,upload_to="images/")
    title_tag=models.CharField(max_length=255,default="")
    author=models.ForeignKey(Profile,on_delete=models.CASCADE)
    caption=RichTextField(blank=True,null=True)
    post_date=models.DateField(auto_now_add=True)
    location=models.CharField(max_length=255,default="")
    no_of_likes=models.IntegerField(default=0)

    def _str_(self):
        return self.title + " | " + str(self.author)

    def get_absolute_url(self):
        return reverse('home')


    def get_owner_pp(self):
        return self.author.profileimg.url


    def profileid(self):
        return self.author.user.id


class Comment(models.Model):

post=models.ForeignKey(Post,related_name="comments",on_delete=models.CASCA
DE)
    name=models.CharField(max_length=255)
    body=models.TextField()
    date_added=models.DateTimeField(auto_now_add=True)


    def _str_(self):
        return '%s - %s' % (self.post.title,self.name)


    def get_absolute_url(self):
        return reverse('home')
class LikePost(models.Model):
    post_id=models.CharField(max_length=500)
    username=models.CharField(max_length=100)
    def _str_(self):
        return self.username
```

APPENDIX – B

SCREENSHOTS

Sample Output

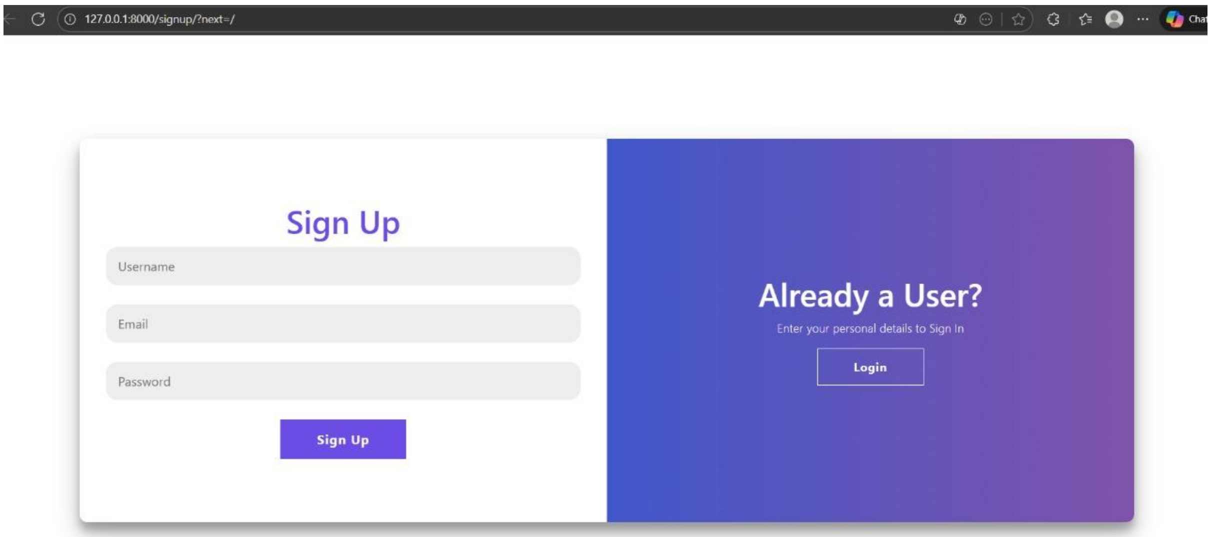

127.0.0.1:8000/signup/?next=/
Sign Up
Username
Email
Password
Sign Up
Already a User?
Enter your personal details to Sign In
Login

Figure A.1: sign up page

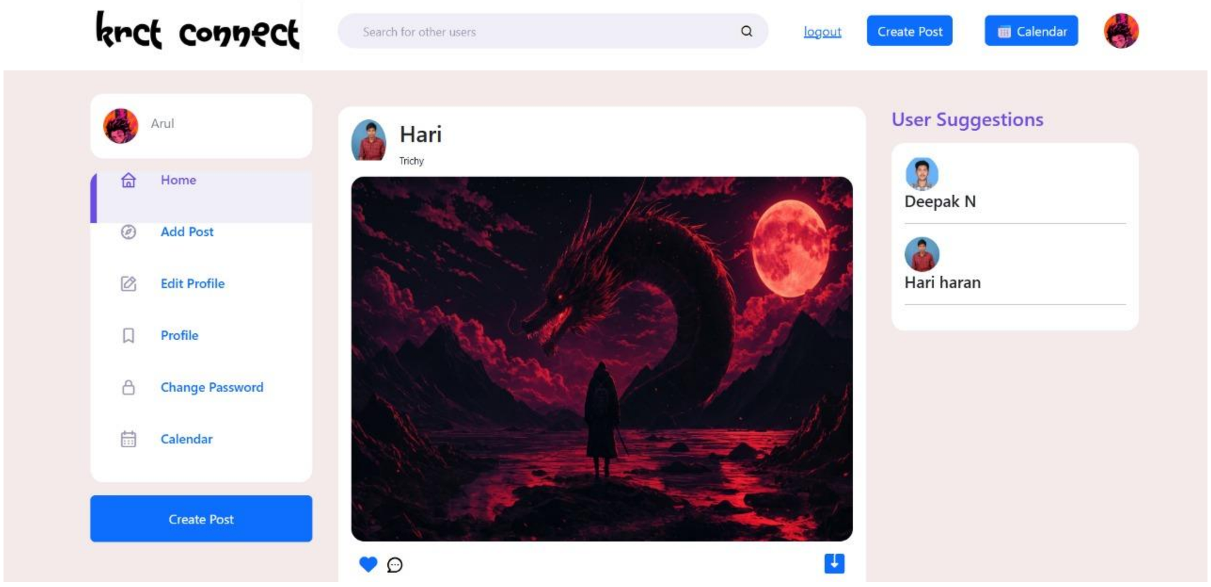
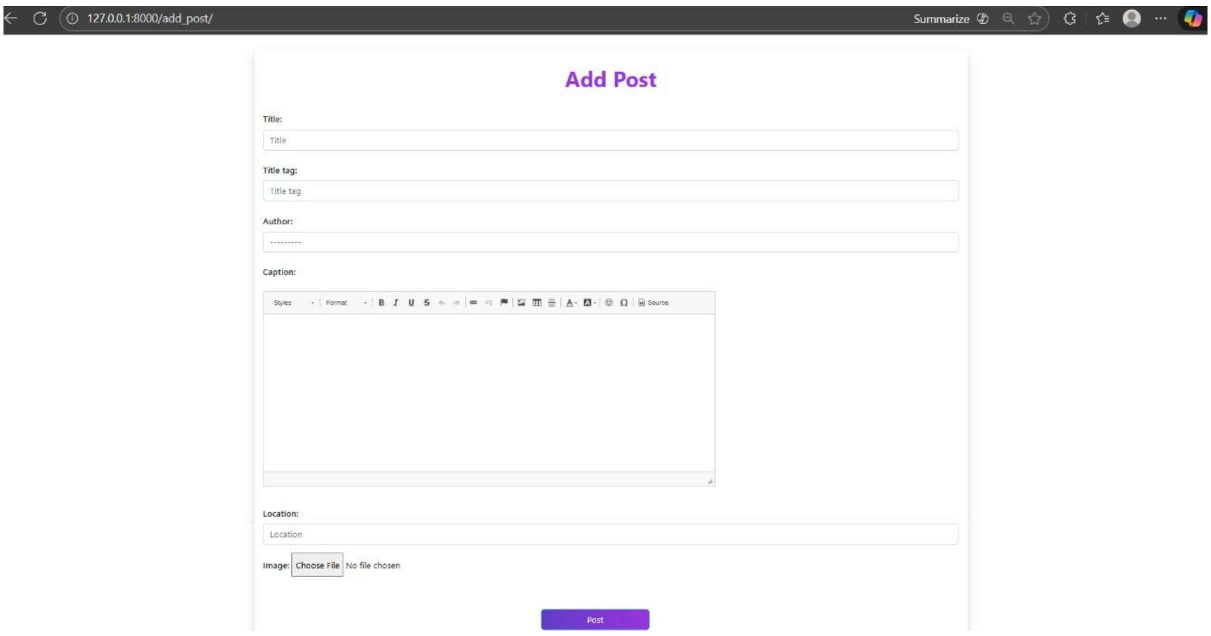


Figure. A.2: home page



Add Post

Title:

Title

Title tag:

Title tag

Author:

Caption:

Location:

Location

Image: No file chosen

Figure A.3: add post page

REFERENCES

1. A. Holovaty and J. Kaplan-Moss, The Django Book, Django Software Foundation, 2009.

2. B. K. Sahu and S. K. Lenka, “Design of a Private Social Networking Platform for Academic Institutions,” IEEE Int. Conf. on Advanced Computing, pp. 540–545, 2020.

3. Goerzen, J. and B. Collins, Foundations of Python Network Programming, Apress, pp. 112–134, 2016.

4. Holovaty, A. and J. Kaplan-Moss, The Django Book, Version 2.0, Django Software Foundation, 2009.

5. Kumar, S. and R. Singh, “Lightweight Web Applications Using SQLite for Educational Environments,” International Journal of Computer Applications, vol. 176, no. 25, pp. 1–5, 2019.

6. Lee, L. and T. Chen, “Social Interaction Systems in Academic Communities: A Web-Based Approach,” International Journal of Educational Technology, vol. 12, no. 3, pp. 33–41, 2020.

7. Naone, E., “Building Secure Web Applications Using Email-Based Authentication,” IEEE Internet Computing, vol. 19, no. 4, pp. 72–78, 2015.

8. Raj, P. and A. Joseph, “Event-Driven Web Applications for College Campus Management,” ACM Journal of Computing and Sustainable Development, vol. 8, no. 2, pp. 44–52, 2021.

9. Sahu, K. and S. K. Lenka, “Design of a Private Social Networking Platform for Academic Institutions,” IEEE International Conference on Advanced Computing, pp. 540–545, 2020.

10. P. Senthil and A. Rao, “Development of a Student-Centric Social Interaction Portal Using Django Framework,” Int. Conf. on Information Technologies, pp. 302–308, 2024.

11. Sharma, K. and V. Gupta, “User Authentication and Secure Access Control in Web Applications Using Domain-Restricted Emails,” IEEE Conference on Information Security and Privacy, pp. 210–216, 2022.

12. Spirin, M. and D. Holubinka, “Responsive User Interface Development Using Bootstrap Framework,” International Conference on Computer Graphics and Design, pp. 148–153, 2018.

13. S. Tejas and R. Vinoth, “Web-Based Peer Networking Systems for Colleges,” Int. J. Advanced Computing, vol. 9, no. 2, pp. 44–52, 2024.

14. S. Warhade et al., “Social App Using Django and Python,” Int. J. Sci. Dev. Res., vol. 7, no. 6, pp. 170–172, 2022.

15. S. Williams, Pro Django, Apress, pp. 65–102, 2018.