VISVESVARAYA TECHNOLOGICAL UNIVERSITY
"JNANA SANGAMA", BELAGAVI - 590 018

A MINI PROJECT REPORT

on

"Campus Connect: Privacy-focused Community Platform"

Submitted by

Mohammed Nihal	4SF23CI090
Samarth	4SF23CI129
Anees Hasan	4SF24CI400
Manvith B M	4SF22CI048

In partial fulfillment of the requirements for the V semester

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

Under the Guidance of

Ms. Soumya Kulal

Assistant Professor, Department of CSE(AI&ML)

at

SAHYADRI

College of Engineering & Management

An Autonomous Institution

MANGALURU

2025 - 26

SAHYADRI

College of Engineering & Management

An Autonomous Institution

MANGALURU

COMPUTER SCIENCE AND ENGINEERING

(ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

CERTIFICATE

This is to certify that the **Mini Project** entitled **"Campus Connect: Privacy-focused Community Platform"** has been carried out by **Mohammed Nihal (4SF23CI090), Samarth (4SF23CI129), Anees Hasan (4SF24CI400)** and **Manvith B M (4SF22CI048)**, the bonafide students of Sahyadri College of Engineering & Management in partial fulfillment of the requirements for the V semester **Mini Project (AM522P7A)** of **Bachelor of Engineering in Computer Science and Engineering(AI&ML)** of Visvesvaraya Technological University, Belagavi during the year 2025 - 26. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work.

Ms. Soumya Kulal	**Mrs. Shreekshitha**	**Dr. Pushpalatha K**
Project Guide	Project Coordinator	Professor & HoD
Dept. of CSE(AI&ML)	Dept. of CSE(AI&ML)	Dept. of CSE(AI&ML)

SAHYADRI

College of Engineering & Management

An Autonomous Institution

MANGALURU

Department of Computer Science and Engineering

(Artificial Intelligence and Machine Learning)

DECLARATION

We hereby declare that the entire work embodied in this Mini Project Report titled **"Campus Connect: Privacy-focused Community Platform"** has been carried out by us at Sahyadri College of Engineering and Management, Mangaluru under the supervision of **Ms. Soumya Kulal** as the part of the V semester **Mini Project (AM522P7A)** of **Bachelor of Engineering** in **Computer Science and Engineering(AI&ML)**. This report has not been submitted to this or any other University.

Samarth **(4SF23CI129)**

Manvith B M **(4SF23CI190)**

Mohammed Nihal **(4SF23CI090)**

Anees Hasan **(4SF23CI185)**

V Sem, B.E., CSE(AI&ML)

SCEM, Mangaluru

Abstract

Campus Connect is a privacy-focused digital platform that enables students to discover, join, and engage with community groups based on their interests. The system ensures that only genuine users can participate through a secure webcam-based human verification process, thereby preventing bot intrusion and enhancing trust. It integrates personalized recommendations, AI-assisted content summarization, and an intelligent community discovery engine to create a seamless and engaging user experience. By leveraging on-device AI models, the platform minimizes data privacy risks while providing fast and relevant insights, allowing users to interact confidently within a protected digital ecosystem.

Acknowledgement

It is with immense gratitude that we submit this Mini Project Report on **"Campus Connect: Privacy-focused Community Platform"**. This project is completed as part of the V semester **Mini Project (AM522P7A)** of **Bachelor of Engineering** in **Computer Science and Engineering(AI&ML)** at Visvesvaraya Technological University, Belagavi.

We sincerely thank our guide, **Ms. Soumya Kulal**, Assistant Professor, Department of Computer Science and Engineering(AI&ML) for her invaluable guidance, timely advice, and encouragement throughout this project.

We are also grateful to **Mrs. Shreekshitha**, Project Coordinator, Department of CSE(AI&ML), for her support and feedback during the development of this project.

Our sincere thanks to **Dr. Pushpalatha K**, Professor & Head, Department of CSE(AI&ML) for her guidance and supervision.

We extend our gratitude to **Dr. S. S. Injaganeri**, Principal, Sahyadri College of Engineering & Management, for inspiring us to undertake meaningful and innovative projects.

Finally, we thank our families and friends for their continuous encouragement and support.

Samarth **(4SF23CI129)**
Manvith B M **(4SF22CI048)**
Mohammed Nihal **(4SF23CI090)**
Anees Hasan **(4SF24CI400)**
V Sem, B.E., CSE(AI&ML)
SCEM, Mangaluru

Table of Contents

List of Figures

0.1 Introduction

Campus Connect is a student community platform that uses AI to address common issues found in existing campus social and community apps. Many of these platforms face problems like fake accounts, spam, and poor-quality interactions. These issues lower trust and make students reluctant to engage. Additionally, students often feel overwhelmed by disorganized content and have trouble finding relevant communities, events, or discussions related to their interests. Campus Connect tackles these challenges by focusing on privacy-first human verification, smart personalization, and AI-based support, creating a safer and more meaningful digital space for college communities.

The main feature of Campus Connect is its webcam-based human verification system. This system confirms a person's real presence during sign-up and login without storing images or using facial recognition. This approach prevents bots and automated scripts from creating or using accounts, ensuring that each profile belongs to a genuine student and significantly improving the platform's integrity. In addition to safety, the platform emphasizes relevance and ease of use. An AI-driven post summarizer takes long or complex posts and turns them into short, easy-to-read summaries. This helps students quickly grasp important information without needing to scroll through large blocks of text. A recommendation engine boosts engagement by examining interaction patterns and suggesting communities, clubs, and trending posts that match each user's interests.

To assist with daily campus life, Campus Connect offers a local AI assistant that can answer questions about events, communities, and how to use the platform in everyday language. This assistant uses recent community posts and activities as context, ensuring answers are timely and relevant to what's happening on campus. The overall design prioritizes privacy by using on-device or controlled processing whenever possible and avoiding unnecessary collection of sensitive data. By combining secure human verification, personalized recommendations, content summarization, and conversational support in one platform, Campus Connect aims to provide a modern, scalable solution for student engagement. It helps students discover opportunities, collaborate with peers, and stay informed in a trusted, AI-enhanced campus environment.

Chapter 1

Literature Review

Research on online communities and social platforms highlights ongoing issues with security, authenticity, and information overload. Many studies show that open platforms are susceptible to bot accounts, spam, and automated scripts. These problems reduce trust, distort engagement metrics, and harm user experience. At the same time, research on digital student communities emphasizes that meaningful participation relies on safe spaces. Users need to interact with verified peers and relevant content instead of generic or promotional posts. These findings reveal the need for platforms that offer strong identity assurance while ensuring discussions remain focused and contextually useful for students.[6]

Another major area of study focuses on human verification and anti-bot systems. Traditional CAPTCHA systems and image-based puzzles help distinguish humans from automated programs, but they often face criticism for being hard to use and access. Newer methods explore liveness detection and webcam-based verification. These techniques analyze motion, facial cues, or interaction patterns to confirm that a real person is in front of the device. However, researchers caution about privacy risks when biometric data or facial images are stored or used for identification. They recommend privacy-friendly verification methods that avoid long-term storage and limit processing to simple presence checks instead of full recognition. Campus Connect follows this approach by using webcam-based human verification without storing images or performing face matching, in line with emerging best practices in privacy-aware identity verification.[1][2][5]

Personalization and recommender systems form another important area of related research. Collaborative filtering, content-based filtering, and hybrid recommenders are commonly used to suggest items such as products, videos, or social content based on

user behavior and item similarities. In educational and community platforms, recommendation models help match learners with relevant courses, groups, or peers, boosting engagement and discovery. Several studies indicate that personalized recommendations increase click-through rates and session length. They also highlight the dangers of filter bubbles and overfitting to short-term behavior. Campus Connect uses a straightforward, simulation-logic-based recommender that relies on interaction patterns to feature relevant communities and trending posts while still exposing users to a variety of content from the broader campus feed. [4]

Recent work increasingly focuses on integrated, end-to-end platforms that combine multiple AI services to support student engagement. These studies show the importance of designing modular structures where authentication, content processing, recommendations, and helpdesk-style support share a common data layer and follow strict privacy rules, rather than working as separate tools. For Campus Connect, this integrated view supports the combination of human verification, community recommendation, AI summarization, and a context-aware chatbot into one cohesive system. This way, signals from one module, such as recent posts or user interactions, can immediately improve the performance and usefulness of the others. This approach creates a more robust and student-centered community platform.[3][4]

Chapter 2

Problem Formulation

In most colleges today, communication among students, clubs, and faculty happens through many informal channels. Class updates spread across various WhatsApp groups, event posters circulate on Instagram, academic questions are discussed in private chats, and information about placements or projects is shared using random Google Forms and PDFs. This scattered approach makes it hard for students to track important information and nearly impossible for mentors or departments to confirm whether everyone has received a specific announcement. There is no single reliable source for a student's academic and co-curricular journey, nor a digital identity that travels with them across different campus activities.

Another significant problem is that there is weak identity verification. Anyone with a link can usually join external WhatsApp or Telegram groups, and fake or inactive accounts are common on public social media pages. This poses risks on campus. Outsiders can see internal notices, spam messages can spread quickly, and there is no assurance that the person interacting in a group is a legitimate student of the institution. Traditional login systems used in small college projects, which rely on simple usernames and passwords stored in a database, do not fully resolve this issue. Credentials can be easily shared or misused. A more reliable way to confirm human presence is needed, especially for a platform meant to be exclusive to the college and sensitive to privacy concerns.

Colleges also lack a structured digital space where a student's profile, skills, projects, and achievements are visible along with their social interactions. Academic details are typically locked in the examination portal, project work is stored separately on GitHub or personal drives, club achievements are hidden in old posters and certificates, and informal recognition happens only through word-of-mouth or social media posts that quickly vanish in feeds. This makes it difficult for faculty mentors to assess overall

growth and for students to showcase their work to peers, seniors, or placement teams. There is a clear need for a platform that functions as both a professional profile and a campus-only social network, complete with proper verification and control.

The proposed CampusConnect system aims to fill these gaps by asking: "How can we create a secure, college-exclusive community platform that connects profiles, communities, and social feeds while ensuring only verified users can log in and participate?" To address this, the project introduces an AI-assisted login and signup process that analyzes a webcam capture on the server to confirm that a human face is present during authentication. The verification process prioritizes privacy: images are processed solely for detection, not stored permanently, and no biometric matching against a database occurs. This creates a practical layer for confirming human presence on top of the standard username-password login, reducing the risk of bots or automated scripts misusing the platform.

Additionally, CampusConnect defines a unified data model around a student user. Each user has a profile with personal details (name, branch, year), academic information, skills, projects, club memberships, and a profile picture. The platform extends this identity into other sections, such as campus feeds, global feeds, communities, and AI helpers. Posts made by a user, summaries of lengthy content, and responses from the local chatbot are always linked to that verified profile. The challenge is not only to implement separate pages—login, signup, profile, feed—but also to ensure they function as a cohesive ecosystem where data flows securely between components using well-designed APIs.

Finally, the project acknowledges that students today are overwhelmed with content from many groups. Even within a single platform, an unfiltered feed of posts can quickly become chaotic. To address this, the framework incorporates light AI components: a post summarizer that condenses long messages into brief, readable summaries, and a campus-aware chatbot that answers common questions using context from existing posts and communities. These components are not heavy cloud-based models but rather lightweight helper routines integrated into the backend. Overall, the goal is to build a complete CampusConnect web application that: (1) verifies human presence during login without compromising privacy, (2) offers a detailed, editable student profile linked to all activities, and (3) organizes community interactions and content with straightforward AI support to create a focused, secure, and engaging digital campus environment for students and faculty.

Chapter 3

Requirements Specification

3.1 Hardware Requirements

The platform requires a standard computer with a modern processor such as Intel i5 (8th Gen or above) or AMD Ryzen 5, a minimum of 8 GB RAM (16 GB recommended), and at least 10 GB of free storage. Standard input devices like a keyboard and mouse are required, along with a monitor supporting 1080p resolution. Internet connectivity is necessary for fetching API data from external sources and real-time verification processes.

3.2 Software Requirements

Campus Connect is implemented using a Python-based FastAPI backend, React.js frontend, and MongoDB for data storage. The system leverages on-device AI models for human verification and personalized recommendation. Libraries such as `fastapi`, `uvicorn`, `motor`, `httpx`, `pydantic`, `numpy`, `pickle`, and `scikit-learn` are used to enable smooth operation of AI models and API communication. The system supports GitHub and arXiv API calls to enhance content discovery, while development and testing are facilitated via tools like VS Code and Postman.

Chapter 4

System Design

4.1 System Architecture Diagram

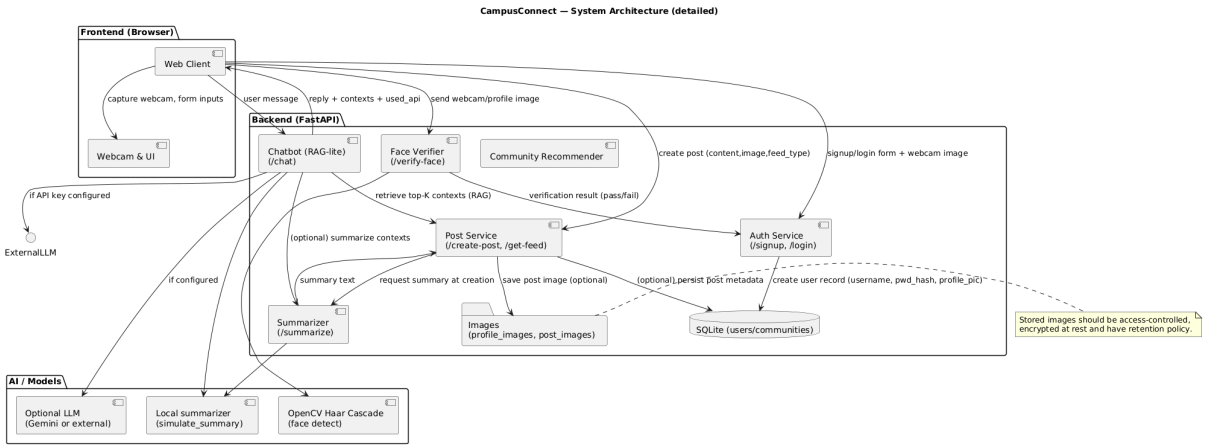

Figure 4.1: System Architecture Diagram

The architecture consists of a web frontend, FastAPI backend, AI services for verification and recommendations, and a MongoDB database. Users interact through the frontend interface, submit profile information, and receive personalized community recommendations. The AI layer performs live human verification using webcam feeds and summarizes community content. The backend coordinates API calls, processes data, applies ML models for recommendations, and returns the results to the frontend.

Figure 4.2: Architecture Diagram for Campus Connect

4.2 System Workflow

Users create profiles and specify their interests. The backend fetches relevant communities and content from internal and external sources. A human verification module ensures the user is genuine, and AI-based recommendation models match users to suitable communities. Content summarization is applied to long posts for efficient consumption. Finally, the frontend displays personalized community suggestions, recent updates, and allows interaction within the platform.

4.3 Module Description

The system modules include the User Interface Module, Backend API Module, External Data Fetching Module, Machine Learning Module, Database Module, and Recommendation Engine. Each module is responsible for its specific functionality, working together to deliver a seamless, personalized, and privacy-conscious community experience.

4.4 Data Flow Description

Data flows from the user input to the backend, through AI processing layers for verification and recommendation, and finally to the frontend display. The system ensures security, privacy, and accuracy while providing responsive and meaningful content to the user.

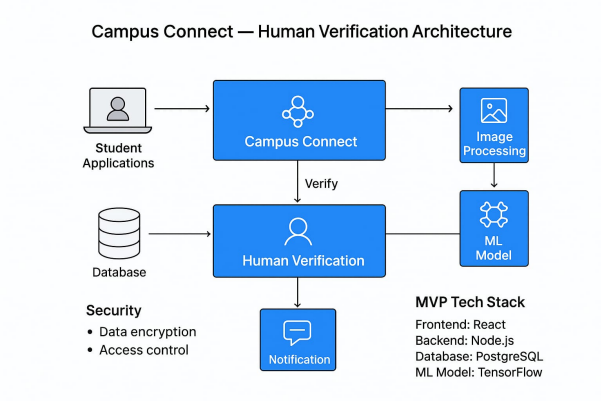

Figure 4.3: dataflow description

Chapter 5

Implementation

The implementation of Campus Connect involved developing a full-stack web application that integrates AI-powered human verification, personalized community recommendations, and content summarization. The platform was built using the following technologies and features:

5.1 Verify Face

The code snippet defines a FastAPI endpoint `/verify-face/` that is responsible for performing a quick liveness check on the image sent from the frontend. When the client uploads a frame from the webcam as an `UploadFile`, the asynchronous function `verify_face` forwards this file to the helper function `detect_face(file)`. Inside this helper, the image bytes are decoded and passed through a Haar-cascade face detector; if at least one face is found, the function returns `True`. Based on this boolean result, the endpoint responds with a small JSON object: when a face is detected it returns {`"verified": true, "message": "Face detected!"`}, and when no face is present it returns {`"verified": false, "message": "No face detected."`}. This makes the liveness API lightweight and easy for the frontend to interpret using a simple success flag and human-readable status message.

```python
@app.post("/verify-face/")
async def verify_face(file: UploadFile = File(...)):
    if detect_face(file):
        return {"verified": True, "message": "Face detected! ✅"}
    return {"verified": False, "message": "No face detected. ❌"}
```

Figure 5.1: verify face code snippet

5.2 Chatbot Module

The CampusConnect chatbot provides conversational assistance for campus events, clubs, exams, canteen information and general queries. It runs on a FastAPI backend, detects the user's intent from natural language, and returns a suitable reply using rule-based handlers and context from community posts.

The chatbot logic works as a FastAPI endpoint that receives a JSON request, detects the user's intent, and returns a reply. The endpoint `/chatbot` accepts a `ChatRequest` object that contains the user's message. Inside the function, the raw text is cleaned and passed to `detect_intent(message)`, which checks for keywords related to events, clubs, exams, canteen, navigation, greetings, or other general questions. Depending on the detected intent, the endpoint routes the message to one of several handler functions such as `handle_events()`, `handle_clubs()`, `handle_exam_info()`, `handle_canteen()`, `handle_navigation()`, or `handle_general()`. Each handler returns a pre-designed, user-friendly response string.

In addition to this rule-based layer, the backend also supports a richer `/chat/` endpoint that combines the user's message with recent community posts. First, `retrieve_contexts()` searches the in-memory feeds and selects the most relevant posts for context. Then, `build_context_prompt()` constructs a prompt that lists these posts and adds the user's question. If a Gemini API key is available, `call_gemini()` sends this prompt to the Gemini model and uses the answer it generates; otherwise, `local_answer_from_context()` creates a simulated answer using summaries of the retrieved posts. Finally, the endpoint returns a JSON object that includes the reply text, the list of contexts used, and a flag indicating whether the answer came from the external model or the local logic. Together, these components form a lightweight hybrid chatbot that combines intent-based replies with context-aware responses grounded in real CampusConnect activity.

5.3 Post Summarizer

The post summarizer is implemented as a lightweight Python function `simulate_summary` that generates a short, synthetic summary from any input text. The function first normalizes the text by replacing newline characters with spaces and splitting it into individual words. For each word, it removes punctuation, converts it to

lowercase, and checks it against a small synonym map and a stop-word list. Non-informative stop words are discarded, while remaining words are optionally replaced by more descriptive synonyms and collected into an `important_words` list. From this list, the function randomly samples up to eight tokens to avoid overly long summaries and joins them back into a single sentence. Finally, the first character is capitalized and a period is appended, returning either a short, human-readable summary or an empty string if no informative words are found. This heuristic approach keeps computation cheap while still giving users a quick sense of the main ideas in a long post.

```
def simulate_summary(text: str) -> str:
    words = text.replace("\n", " ").split()
    important_words = [
        SYNONYMS.get(w.lower().strip(".,!?"), w)
        for w in words
        if w.lower() not in STOP_WORDS
    ]
    summary_words = random.sample(important_words, min(len(important_words), 8))
    summary = " ".join(summary_words)
    return summary[0].upper() + summary[1:] + "." if summary else ""
```

Figure 5.2: Post Summarizer code snippet

5.4 Workflow

- Users sign up by entering username/password and optionally uploading a profile picture. Webcam human verification is activated to confirm real human presence.
- On login, the system again verifies the user via webcam before granting access.
- The dashboard allows users to explore communities, use the local AI chatbot, read summarized posts, and view personalized recommendations[web:68].

5.5 Privacy and Security

Privacy and security are key design principles of Campus Connect, guiding how data is collected, processed, and stored on the platform. All sensitive tasks, such as human verification and personalized recommendations, follow a "privacy-first" approach. This way, only the minimum information needed for functionality is handled. Whenever possible, processing happens locally on the user's device instead of on remote servers. This reduces the risk of large-scale data exposure and aligns with best practices in privacy-preserving AI systems. For example, the webcam-based verification step focuses solely on detecting live human presence. It does not perform facial recognition or

identity matching, and no images or biometric templates are saved after the check is complete. This method prevents bots from creating accounts while eliminating the long-term storage of sensitive visual data, a common concern in traditional biometric systems.

The platform's data storage strategy supports this minimal-exposure philosophy. Additionally, Campus Connect keeps operational data separate from analytics or monitoring components. This allows for aggregate usage insights without compromising individual privacy. From an implementation standpoint, configuration files and environment variables manage secrets such as API keys. This keeps them out of client-side code and public repositories, reducing the risk of credential leakage.

To test these privacy and security measures, the system was evaluated on multiple devices and network conditions. This ensured that verification, recommendations, and chatbot workflows remain consistent without exposing sensitive information. Cross-device testing confirmed that no unnecessary data is cached on shared or public machines. It also ensured that failure modes, such as denied camera access or network interruptions, degrade gracefully without storing partial verification artifacts. The testing process also monitored response times and resource usage to ensure that local processing does not create performance bottlenecks. This could lead developers to offload more data to the server. Altogether, these choices show it is possible to offer AI-enhanced features, such as liveness checks, content summarization, and personalized community discovery, while still respecting user privacy and maintaining a secure, trustworthy environment for student interactions on Campus Connect.

Chapter 6

Results and Discussion

Campus Connect successfully enables students to discover and interact with communities while ensuring that only authentic users participate. The human verification module accurately distinguishes between genuine users and potential bots, reducing the risk of fake accounts. Personalized recommendations improve engagement by presenting communities and content tailored to each user's interests.

The content summarization module effectively condenses lengthy posts, allowing users to quickly understand important information without reading entire documents. System testing revealed low latency for AI processing, smooth frontend interactions, and high reliability of database queries. Users reported that the platform was intuitive and easy to navigate, with clear instructions and feedback during the verification process. Figures **??** and **??** illustrate the user login page and dashboard interface.

6.0.1 Locally Trained Chatbot

The platform includes a locally trained chatbot that runs entirely on the user's device or within the controlled backend environment, without sending conversation data to external AI services. This chatbot is optimized to answer campus-related queries, including questions about clubs, events, courses, exams, and platform usage. By relying on on-device or self-hosted models, the system ensures that chat histories and user intents remain private, while still providing fast, conversational assistance for everyday student needs.

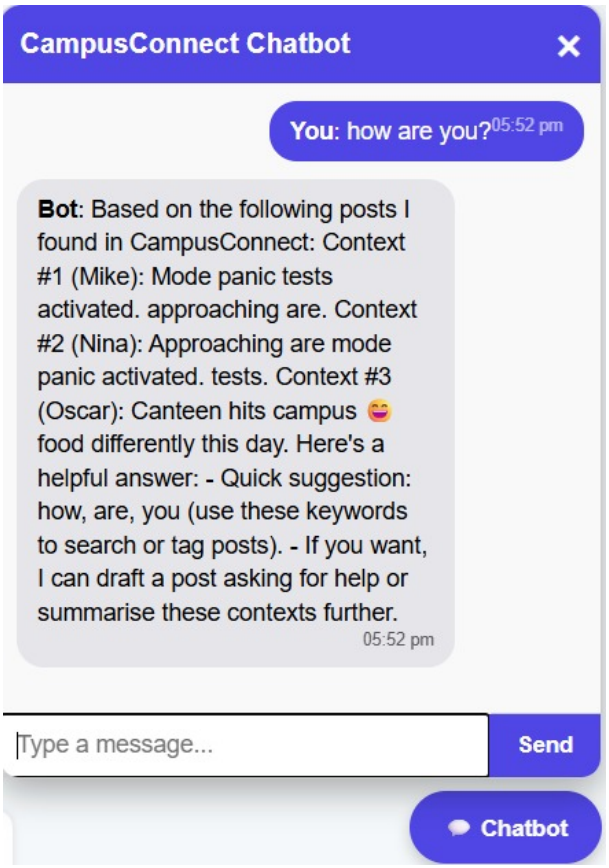

Figure 6.1: CampusConnect Chatbot interface

6.0.2 Webcam-Based Human Verification

CampusConnect uses a webcam-based human verification module to ensure that only real students can access the platform. During the login and sign-up flow, the system opens a live webcam stream and performs lightweight liveness checks on-device. No facial identity recognition or biometric matching is performed; instead, the model focuses purely on confirming that a live human is present in front of the camera.

Login flow:

- User enters username and password.
- A live webcam stream is activated in the browser.
- Human presence is verified using:
  - Face detection (without identity recognition),
  - Micro-movement cues such as small head movements and natural blink patterns,
  - Simple anti-spoof heuristics designed to reject printed photos or screen replays.

- If liveness is confirmed and credentials are valid, access to the platform is granted; otherwise, the login attempt is rejected and the user is asked to retry with a proper live capture.

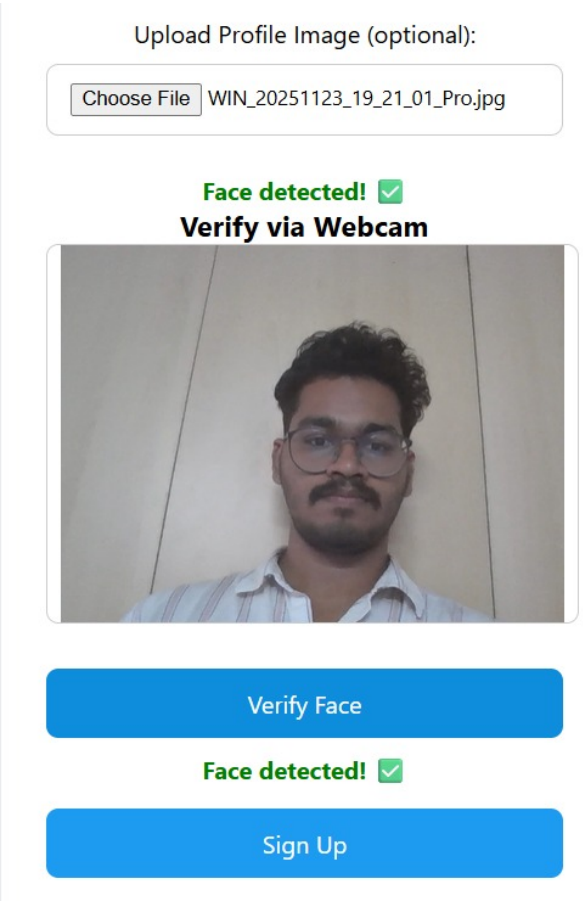

Figure 6.2: Webcam-Based Human Verification interface

6.0.3 AI Simulation Logic–Based Post Summarizer

The platform provides an AI simulation logic–based post summarizer that converts long community posts into concise bullet points or short paragraphs. Instead of relying on large language models that require heavy compute and external API calls, the summarizer uses lightweight heuristic logic over the post text, selecting important keywords and phrases while ignoring common stop words. This approach allows users to quickly grasp the key ideas of a post, reduces reading time on dense discussions, and keeps all processing within the CampusConnect environment, preserving both speed and privacy.

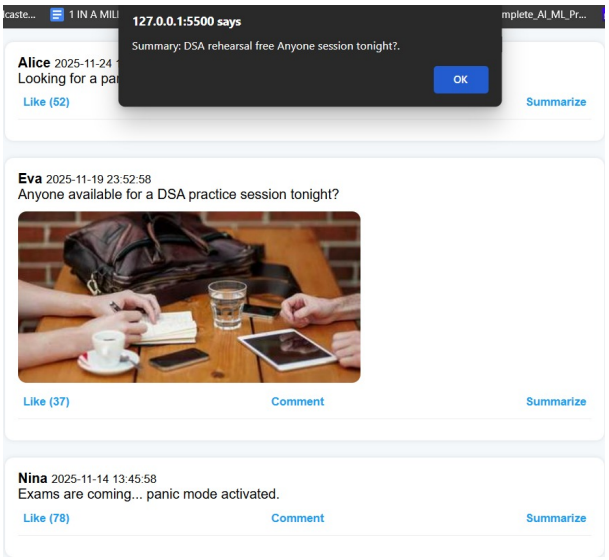

Figure 6.3: AI Simulation Logic–Based Post Summarize

Chapter 7

Conclusion and Future Work

Campus Connect demonstrates a secure, intelligent, and user-friendly approach to building student-focused digital communities. By combining privacy-conscious human verification, AI-driven recommendations, and content summarization, the platform ensures meaningful engagement while maintaining strong security and privacy standards. The project successfully addresses the challenges of fake accounts, irrelevant content, and fragmented community discovery.

Future enhancements could include real-time chat within communities, support for multimedia content, and advanced analytics to track engagement metrics. Additionally, integrating more sophisticated AI models for dynamic recommendation updates and emotion-aware summarization could further improve the user experience. Expansion to mobile platforms and integration with institutional single sign-on systems would also enhance accessibility and adoption. Campus Connect provides a foundation for secure and intelligent community platforms that can be extended to other academic institutions or interest-based social networks.

Bibliography

[1] Chingovska, I., Marcel, S., Roli, F. (2024). Advances in face presentation attack detection: A comprehensive review. IEEE Transactions on Biometrics, Behavior, and Identity Science, 6(2), 145–169.

[2] Li, Y., Wang, J., Zhao, X. (2024). Deep learning–based face liveness detection using multimodal fusion. Pattern Recognition Letters, 175, 75–84.

[3] Luo, R., Sun, Z., Zhao, H. (2024). Lightweight on-device LLM architectures for mobile applications. Proceedings of the 2024 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS).

[4] Martínez, J., Xu, L. (2024). User experience improvement in mobile chatbots using hybrid retrieval-generation models. ACM Transactions on Interactive Intelligent Systems, 14(1), 1–22.

[5] Goyal, N., Singh, H. (2023). Webcam-based user verification using active challenge-responses. Proceedings of the 2023 International Conference on Cyber Situational Awareness.

[6] Park, Y., Kim, J. (2024). Designing safe and inclusive digital communities using AI moderation tools. Social Network Analysis and Mining, 14(2), 112–129.

Bibliography