Tribhuvan University

Institute of Science and Technology

A Final Year Project Proposal

On

"CampusHub: Secure andInteractive Digital Platform"

In partial fulfillment of the requirements for the Bachelor's Degree in Computer Science and Information Technology
(B.Sc. CSIT)

Submitted to:
Butwal Multiple Campus
Department of Computer Science and Information Technology
Butwal, Rupandehi

Submitted by:

Arjun Khatri(25106/76)
Ghanashyam Khanal(25126/76)
Rajendra Pantha(25150/76)

Table of Content

1. Introduction

Assignments serve as valuable tools for tracking the progress of both students and teachers. They allow students to apply and test their understanding, while enabling teachers to assess the effectiveness of their lessons. Our web based applications aids teachers in scheduling tasks and assigning them to the students. Additionally, it provides students with deadlines, and once assignments are submitted, teachers evaluate the work. Our project also have features to start messaging for this users should get connected to server where they can do group and private message and also can share files. [1]

Our application includes an assignment tracker and provides quick notifications. It allows users to send text messages and share various resources, including files and documents, within messages (both in group and private). We have paid close attention to user privacy and security,implementing encryption to ensure that their conversations remain private. This chat application is develop using MERN stack for database, routing, frontend and backend respectively and Socket.io for real time data communication. This project represents our commitment to enhance the intra-collegecommunication in network.

Our web-based, real-time application enables efficient collaboration between professors and students. It provides separate dashboards for both professors and students. The professor's dashboard includes features such as a profile, notifications, classes, and messages. In contrast, the student's dashboard comprises a profile, notifications, subjects and messages. This platform offers a well-organized and convenient space for seamless interaction.

2. Problem Statement

The necessity for a specialized college application arises due to challenges associated with email communication and in-person meetings. Such a web-based application is designed to simplify assignment tracking, enhance communication between students and professor, and provide convenience for sharing class materials and staying informed about college events. Some of problems that may occur are:

- **Limited Access to Educational Resources:** Without a dedicated platform, students and teachers may struggle to access and share educational materials efficiently, limiting the overall learning experience.
- **Disorganized Information:** Lack of a structured system may result in disorganized information storage, making it difficult for students to find study materials, assignments, and resources when needed.
- **Lack of Feedback Mechanism:** Students and teachers may lack a structured way to provide feedback on the learning process, hindering continuous improvement.
- **Integration Hassles:** The absence of a unified platform may require students and teachers to juggle multiple tools, leading to inefficiencies and confusion.

3. Objectives

Some of our projects objectives are as follow:

- Develop a specialized web-based college application.
- Establish a feedback mechanism for continuous improvement.
- Implement structured information storage and retrieval.
- To Transfer different file format over the system.
- Assignment tracking of students by professor.
- To make users get connected at anytime from anywhere. [2]

4. Methodology

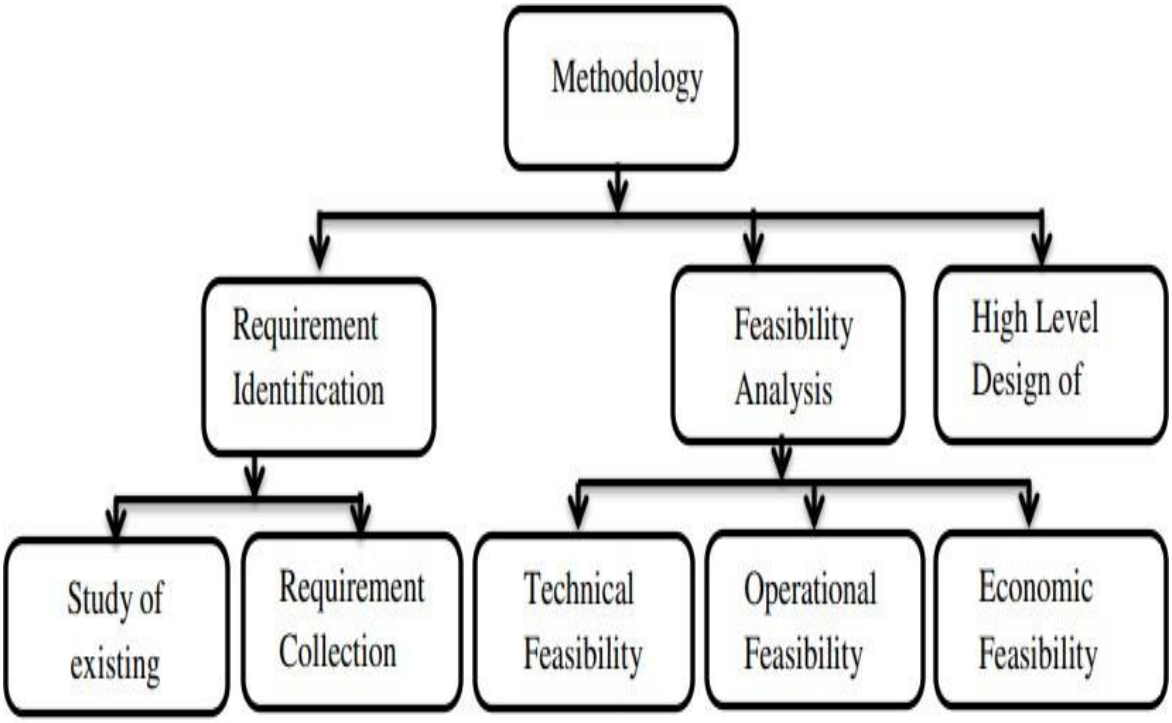

Figure 1: Research Methodology

4.1 Requirement Analysis

- **Functional requirement**
  - User Registration: User must be able to register for the application through a valid email, username and password. On opening the application user must be prompted to register their email, username and password. If user skips this step application should close.
  - Send Messages: User should be able to send instant messages to any contact on his/her chat application contact based.
  - Assignment : Professor must be able to create assignment and maintain record of student and student must be able to submit assignment.
  - Broadcast message: User should be able to create a group of contacts. User should be able to broadcast message to these groups.

- **Non Functional requirements**

  - Privacy: Messages shared between users should be encrypted to maintain privacy.

  - Performance: Application must be lightweight and must sent messages instantly.

  - Interface**:** Application interface must be simple in design and interactive layout. [2]

4.2 Requirement Identification

4.2.1 Study of Existing System / Literature Review

This research investigated the structure and design of a real time collaboration programmed thatallows many users to send and receive coordinate documents through a web browser.

Chat apps (and their subsets, chat rooms) bring to remembrance images of the 1990s, with its dial-up internet and classic sitcoms, however, commercial chat apps date back to the 1980s. CompuServe released CB Simulator in 1980, and 1985 brought the launch of Commodore's Quantum Link (also known as Q-Link). An online service, it allowed multiuser chat, email, file sharing, and games. As the first decade of the new millennium closed, an enterprise chat app renaissance slowly began. Though email had been widely used for the previous twenty

years, companies soon began looking for a better way to communicate quickly; email-based workflows are slower and do not allow for many business functions that are now critical to work, like screensharing or video calls. Some one-to-one chat applications existed, like GChat and Outlook Messenger, but group messaging applications had yet to take off. [4]

4.2.2 Tools to be used

Following are the tools to be used for the development of this project:

- Front-End
  - HTML
  - CSS
  - JS
  - React JS
  - Tailwind CSS
- Back-End
  - Node.js
  - Socket.io
- Database
  - Mongo Db

4.3 Feasibility Study

4.3.1 Technical Feasibility:

The proposed system can be easily maintained and repaired; technically, the system will be powerful to be applied by low skilled users as much as possible. There is no need for the developer involvement in almost all implementation of the entire system. It is easily accessible by the people who can easily understand natural languages.

4.3.2 Operational Feasibility:

It will be operational for 24hrs a day and user can access necessary resources available in the chat application at any needed time. The system gives better user interface registration form and storage of user information, easy operating, deletion and modification etc.

4.3.3 Economical Feasibility:

There is no need of extra hardware components and paid software to build the system. Cost of Software to be acquired to build and run the product is a onetime cost. Its maintenance cost is cheaper as compared to other.

4.3.4 Schedule (Gantt chart):

Activities	1W	2W	3W	4W	5W	6W	7W	8W	9W	10W	11W	12W
Planning	■	■	■									
Analysis		■	■	■								
Design			■	■	■	■	■					
Implementation				■	■	■	■	■	■	■		
Testing								■	■	■	■	
Documentation	■	■	■	■	■	■	■	■	■	■	■	■
Review												■
Presentation												*

Figure1. Gantt chart

4.4 High level design of system

Our system is used to maintain conversation and activities between teachers andstudent. In this project there are mainly 2 modules:

- Professors
- Student

Professor: Each Professor is given a user id and password through which they are able to login to the application. Each professor has following option :

- Change password and update personal details.
- Creating and modifying assignment and adding comment to assignments.
- Monitoring student activities.
- Privilege of joining private chat.

Students: Each Student is given a user id and password through which they are able to login to the application. Each Student has following options:

- Change password and update personal details.
- Submit assignment.
- Privilege of joining private chat.

4.3.1 Use case Diagram

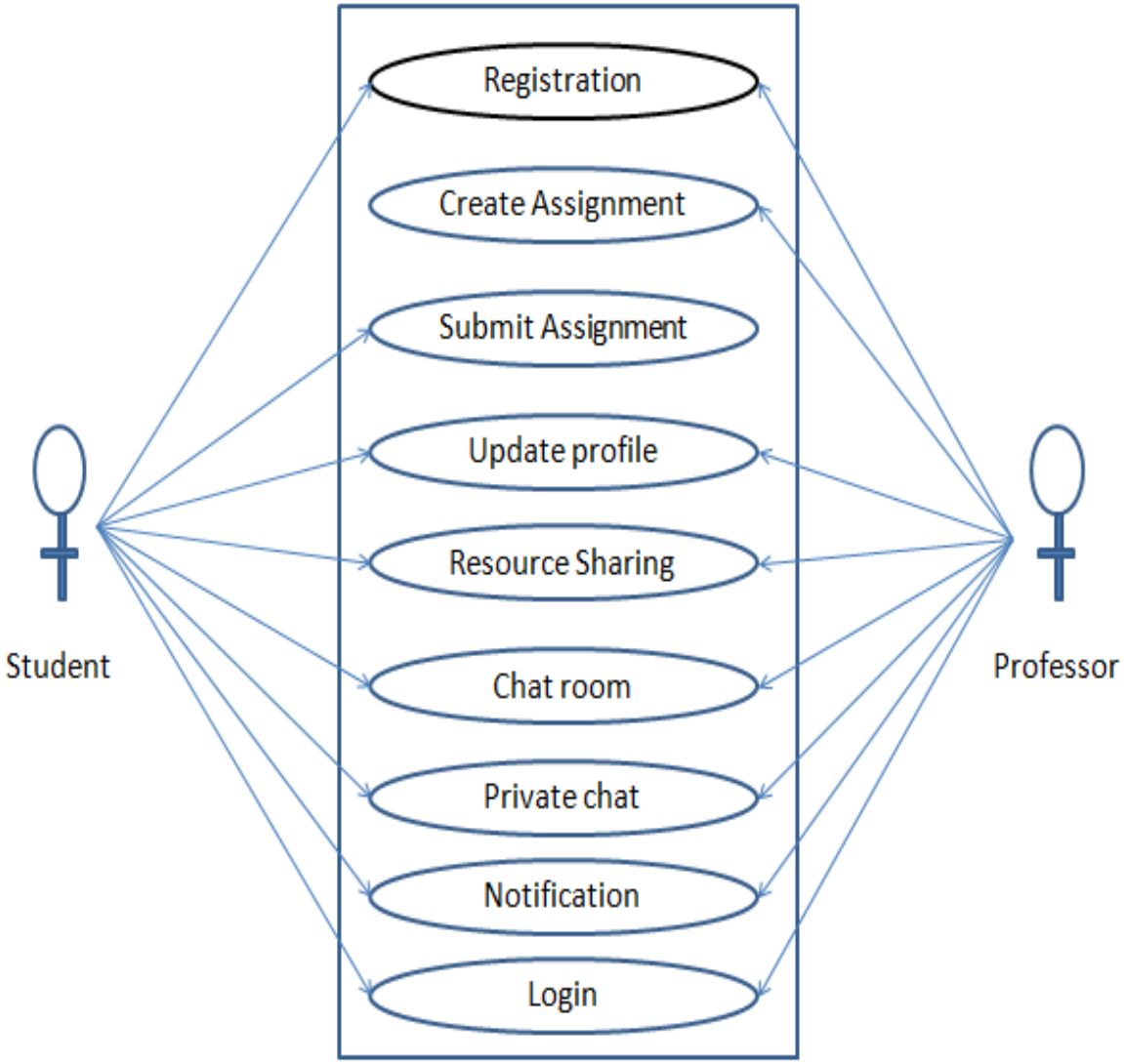

Figure: Use case diagram

4.3.2 Flow chart of the system

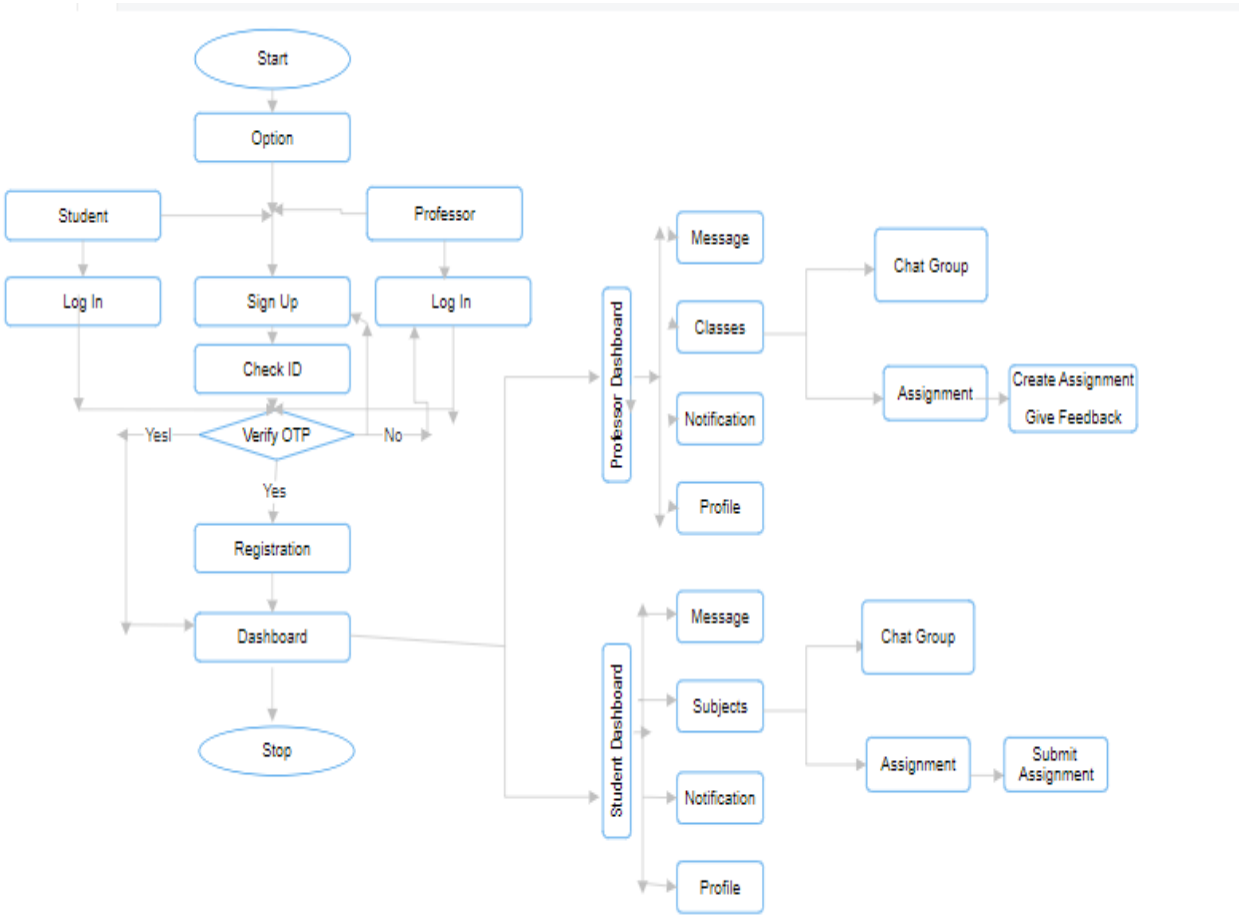

Figure: Flow chart

4.4.3 Algorithm Used

➢ **Caesar Cipher algorithm for encryption and decryption**

The Caesar Cipher is a simple substitution cipher that encrypts and decrypts messages by shifting each letter in the alphabet a fixed number of positions forward or backward.

Encryption of a letter 'x' by a shift value 'n' can be described mathematically as:$E_n(x)=(x+n) \mod 26$.

```
Function encrypt(text, shift):

result = ""

For each character in text:

    if character is a letter:
        if character is uppercase:
            encrypted_char = (character + shift - 'A') % 26 + 'A'
        else:
            encrypted_char = (character + shift - 'a') % 26 + 'a'
    else:
        encrypted_char = character

    append encrypted_char to result

  Return result
```

Decryption of a letter 'y' that was encrypted with a shift value 'n,' you use the formula: $D_n(x) = (xi - n) \mod 26$

where n is fixed key used for encryption and decryption and ‘x’ can be the number given the alphabet or its ASCII value

```
Function decrypt(ciphertext, shift):
   result = ""
 For each character in ciphertext:
    if character is a letter:
        if character is uppercase:
            decrypted_char = (character - shift - 'A' + 26) % 26 + 'A'
        else:
            decrypted_char = (character - shift - 'a' + 26) % 26 + 'a'
    else:
        decrypted_char = character

    append decrypted_char to result

  Return result
```

- **Knuth–Morris–Pratt algorithm**

Knuth–Morris–Pratt algorithm (or KMP algorithm) is a string-searching algorithm that searches for occurrences of a "word" W within a main "text string" S by employing the observation that when a mismatch occurs, the word itself embodies sufficient information to determine where the next match could begin, thus bypassing re-examination of previously matched characters.

- Text Searching: KMP is widely used for efficient text searching, allowing you to find specific words or phrases within large documents or datasets.

- String Matching: It's valuable for determining if a particular pattern or substring exists within a set of strings or a database.

- Compiler Design: In compiler construction, KMP helps in lexical analysis to identify keywords and symbols in programming languages efficiently.

KMP-MATCHER (T, P)

1. n ← length [T]
2. m ← length [P]
3. Π← COMPUTE-PREFIX-FUNCTION (P)
4. q ← 0 // numbers of characters matched
5. for i ← 1 to n // scan S from left to right
6. do while q > 0 and P [q + 1] ≠ T [i]
7. do q ← Π [q] // next character does not match
8. If P [q + 1] = T [i]
9. then q ← q + 1 // next character matches
10. If q = m // is all of p matched?
11. then print "Pattern occurs with shift" i - m
12. q ← Π [q] // look for the next match

4. Expected Output

- **Personal and Group Communication**: Users can engage in both one-on-one and group conversations, promoting effective communication among individuals and teams

- **Ease and Speed of Communication**: Users experience a user-friendly platform that enables quick and effortless communication, improving overall efficiency.

- **24/7 Connectivity:** The system ensures users can connect with each other anytime and from anywhere.

- **Assignment tracking:** It assists teachers in scheduling tasks, assigning them to the students, and providing quick notifications.

References

[1] A. Mohamad, A. Muhammed and M. Man, "A Secure Chat Application Based on Pure Peer-to-Peer," *Journal of Computer Science,* vol. 11, no. 5, pp. 723-729, 2015.

[2] G. R and K. C, "Multi-User Chat Application," 07 2020. [Online]. Available: https://www.ijeat.org/wp-content/uploads/papers/v9i5/E9578069520.pdf. [Accessed 28 09 2023].

[3] "WEB BASED CHAT APPLICATION," 11 2019. [Online]. Available: https://www.studocu.com/in/document/north-east-adventist-university/electronics-and-communication/web-based-chat-application-minor-project/28140907?fbclid=IwAR07lDQiV9z_6wOSseyNzxO10zCffAwlA2X5SsrauOZBKZbbrqocPWgefGI. [Accessed 09 2023].

[4] N. Eric, "Design and Implementation of a Real Time Chat Application," 12 2020. [Online]. Available: https://portal.bazeuniversity.edu.ng/student/assets/thesis/20210215120658149063642.pdf?fbclid=IwAR0JKrvJRmdVgmUGDkQXx7WbXp6x5HpGazYtx9vryhQuJvZsPtJaFXkcWa4. [Accessed 28 09 2023].

[5] "SceinceDirect," 2023. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/caesar-cipher#:~:text=The%20Caesar%20cipher%20is%20based,shifts%20in%20the%20opposite%20direction.. [Accessed 28 09 2023].