# SOCIAL CHAT SYSTEM FOR DEVELOPMENT OF RELATIONSHIP & SOCIAL SKILLS

Kailashwaran R.A.K.K

IT21118654

B.Sc. (Hons) Degree in Information Technology

Department of Information Technology
Sri Lanka Institute of Information Technology
Sri Lanka

August 2024

SOCIAL CHAT SYSTEM FOR DEVELOPMENT OF RELATIONSHIP & SOCIAL SKILLS

# SOCIAL CHAT SYSTEM FOR DEVELOPMENT OF RELATIONSHIP & SOCIAL SKILLS

Kailashwaran R.A.K.K

IT21118654

B.Sc. (Hons) Degree in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology
Sri Lanka

August 2024

SOCIAL CHAT SYSTEM FOR DEVELOPMENT OF RELATIONSHIP & SOCIAL SKILLS

# DECLARATION

I declare that this is my own work and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

| **Name** | **Student ID** | **Signature** |
|---|---|---|
| Kailashawaran R.A.K.K | IT21118654 |  |

**TABLE OF CONTENTS**

## LIST OF FIGURES

## Abstract

Numerous difficulties that university students frequently face can have a major negative influence on both their general well-being and academic achievement. Relationship problems, Academic challenges, financial strains, and social isolation stand out as significant variables affecting school results. This abstract investigates how these elements are related to one another and how their combined impact affects students' academic performance. Relationship problems, including disputes with friends, and family, can cause emotional anguish and impair focus, which can negatively impact academic achievement and engagement. These interpersonal disputes might worsen academic difficulties by lowering motivation and raising absenteeism.

Academic struggles themselves, including difficulty understanding course material, poor study habits, and ineffective time management, directly impair academic success. These struggles can generate feelings of incompetence and anxiety which potentially leads to a negative feedback loop where diminished confidence further deteriorates performance. Financial burdens pose another significant obstacle such as students grappling with economic hardships often face additional stress such as the need to work part-time jobs, leading to reduced study time and increased fatigue. Financial insecurity can also limit access to essential educational resources and opportunities widening the achievement gap among students from different socioeconomic backgrounds.

Academic results can be significantly impacted by social isolation, which can be brought on by a lack of support systems, cultural differences, or mental health concerns. Isolation can cause depressive and lonely sentiments, which lower interest in extracurricular and academic activities. Furthermore, the development of critical social and professional skills as well as collaborative learning might be hampered by a lack of peer support. These elements don't always conflict with one another and frequently work in concert to exacerbate their negative effects on academic achievement. For example, social isolation can be exacerbated by financial stress and interpersonal connections may be strained by academic difficulties. Facing the complex nature of these issues is essential to creating all-encompassing support structures in universities.

Implementing solutions such as counselling services, financial aids, academic support workshops and community-building activities can help to minimize these issues. Addressing these factors is essential to enhance academic success and promote the overall well-being of university students, fostering a more equitable and supportive educational environment.



**Keywords: Social interaction, relationship problems, university students, chatbot, social points, React, Firebase, Zustand**

# 1. Introduction

University life is a crucial time for growth and development, marked by the quest for academic success and the creation of deep connections. Notwithstanding the prospects for individual and academic progress, students frequently face noteworthy obstacles in effectively managing relationships, particularly those about family and friends. New obligations, greater freedom, and the need to strike a balance between social contacts and academic demands are all part of the adjustment to university life. These demands may put a strain on current friendships and familial ties, resulting in misunderstandings, arguments and feelings of loneliness. Relationship difficulties of this nature have the potential to significantly impact students' emotional health and in their academic achievement.

For students to feel stable and supported emotionally, family relationships are essential. However, there may be conflict and communication hurdles between students and their families as a result of the physical distance from home and the move towards independence. As students learn to negotiate their new roles and duties, misunderstandings and unmet expectations may occur. This could cause emotions of shame, stress, and diversion from academic goals. Friendships which are crucial for a sense of community and social support—may also suffer during this period of change. Social isolation and a decline in motivation might result from the difficulties of making new acquaintances at a university setting, keeping up long-distance friendships from home, and juggling a demanding academic schedule with extracurricular activities.

The difficulties in these relationships may compound to affect the academic success of the students. Open Conflicts with family and friends can cause emotional anguish that might impair one's ability to concentrate, engage in academic activities, and perform well academically overall. Poor academic results might result from student's inability to concentrate on their studies, participate in group projects or ask for assistance when necessary. Relationship problems can cause tension and anxiety which can worsen depressive and lonely feelings which make it harder for children to do well in school.

This study offers a social chat program as a solution to these problems. It helps college students manage and strengthen their relationships, which in turn improves their academic performance. Through this app Users may have meaningful conversations with their peers, get feedback on how they connect with others and improve their ability to communicate and make relationships. The technology provides students with real-time feedback on their social interactions, enabling them to grade each other on attributes like friendliness, open-mindedness, and communication skills. This helps students identify their areas of strength and growth. With the help of this feedback system, students may be more proactive in enhancing their relationships and creating a stronger, more cohesive social network.

Moreover, the app includes features that promote user safety and privacy, such as a blocking mechanism to prevent unwanted communication. These features are essential in creating a secure environment where students can feel confident in engaging with others and seeking feedback without fear of negative consequences. By addressing the social and emotional challenges associated with relationship struggles this app is aims to minimize the impact on student's academic performance. Improved social interactions and stronger relationships can lead to a more

positive academic experience, where students are better equipped to manage stress, seek support, and stay focused on their academic goals.

In addition to improving students' interpersonal skills and connections, this method helps them succeed academically by offering a disciplined and encouraging environment for personal development. Through guiding users through the intricacies of college life, the app provides an invaluable resource for enhancing students' social and academic performance. This study emphasises how crucial it is to address interpersonal issues as a critical component of students' academic achievement and how technology can offer creative solutions to these problems.

## 1.1 1Background

Many obstacles that Sri Lankan university students must overcome can have a big impact on both their general well-being and academic achievement. Relationship issues, scholastic hurdles, financial strains, and social isolation are among these concerns that significantly affect students' academic performance. To support students and enhance their educational experiences, initiatives that take these elements into account must be developed. In Sri Lanka, relationship problems are a prevalent problem among university students, especially those that include friends and family. Students may experience severe stress due to cultural standards, familial expectations and societal pressures. Emotional anguish resulting from disputes with friends or family can impair a student's capacity to concentrate on their studies.

This psychological crisis frequently shows up as a decline in focus, motivation, and academic engagement, which eventually leads to poor performance in university, another significant issue affecting university students in Sri Lanka is academic challenges. When students move from school to university they have to adjust to more demanding coursework and independent study habits, which can be particularly difficult. Many students have difficulty managing their time, comprehending difficult material and achieving the rigorous standards of schoolwork at the university level. The academic experience of pupils from non-English speaking backgrounds is further complicated by language obstacles. Students may become frustrated and feel inadequate as a result of these academic setbacks, which may deter them from giving their studies their all.

Many university students in Sri Lanka particularly those who from low-income families, are quite worried about their financial commitments. The costs of living expenses and Educational expenses can put students and their families in a difficult financial situation. Many students must work part-time jobs in order to make ends meet which may reduce their productivity and study time. Burnout can result from the stress of balancing the demands of academic work with financial constraints which can negatively impact academic progress. Students who struggle financially may also find it more challenging to have access to the course materials, the internet and study resources that are crucial to their academic success.

Another important aspect that significantly influences Sri Lankan university student's academic performance is social isolation. In order to attend college, many students relocate away from their families, frequently leaving behind their support systems. This might cause students to feel alone

and isolated, especially those who come from under-represented or rural regions. Language hurdles, cultural gaps, and the competitive aspect of academic life can all worsen social isolation. Students who lack strong social networks may find it difficult to get the academic support they require, which could result in a lack of interest in their studies and poorer academic achievement. These problems are linked together, frequently making them worse and weaving a complicated web of problems that can negatively affect children' academic performance.

For instance, social isolation may be exacerbated by financial stress, and family and friend connections may be strained by scholastic difficulties. Understanding how these elements are related to one another is essential to creating all-encompassing support systems at Sri Lankan colleges. In order to tackle these problems, it's critical to put in place focused interventions including financial aid, academic support services, counselling, and community-building activities for students. Teachers and legislators can better assist Sri Lankan university students in attaining both academic success and personal well-being by knowing the history and context of these issues.

## Literature Survey

Students at universities face a complicated interaction of social, economic, and psychological issues that affect their academic achievement. The effects of marital difficulties, academic difficulties, financial strains, and social isolation on students' academic outcomes have been well studied in the past. This literature survey reviews the key findings related to these factors, with a specific focus on the context of Sri Lankan university students.

**Relationship challenges:** It has been extensively shown that relationship challenges, particularly those involving family and peer interactions, have an impact on academic achievement. Research indicates that disputes with family members might result in elevated levels of tension and anxiety, hence adversely influencing academic performance and concentration (Ghafournia, 2014). The educational experiences of students in Sri Lanka are significantly shaped by cultural norms and family expectations. High parental expectations might result in anxiety and stress, which can negatively impact academic achievement, according to Perera and Kathriarachchi (2020). Furthermore, peer relationships are essential for creating a sense of belonging and offering emotional support, both of which are critical for academic performance (Wentzel, 2017).

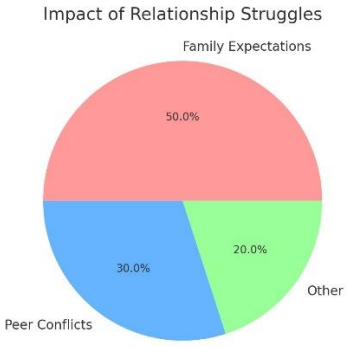



Figure 1. Impact of Relationship Struggles

**Academic problems:** University students experience a number of well-documented academic problems, such as learning how to manage their time, comprehend the content, and adjust to independent study. Academic integration is crucial for guaranteeing student achievement and retention, according to Tinto (1993). Studies unique to Sri Lanka emphasise the difficulties caused by inadequate academic preparation and language obstacles. Students from non-English speaking backgrounds frequently struggle with comprehension, which lowers their academic achievement, according to Fernando and Karunaratne (2015). Additionally, the high stakes of national exams combined with Sri Lanka's competitive academic environment can make students feel more stressed and anxious (Jayawardena & Kumari, 2019).

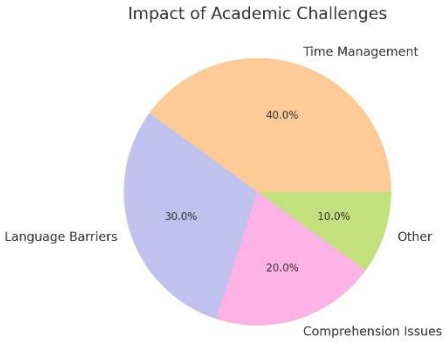



Figure 2. Impact of Academic Challenges

**Financial Burdens:** Several studies have demonstrated that influence of financial stress on students' capacity to achieve academically making it a strong predictor of academic achievement. According to Roberts et al. (2000), students who are having financial issues frequently end up taking up part-time jobs, which might interfere with their study time and increase their risk of weariness and burnout. Low-income students in Sri Lanka have additional challenges due to the high cost of life and education. According to Perera (2018), one of the main causes of Sri Lankan university students' high dropout rates is financial hardship. Furthermore, access to educational resources like textbooks and online materials may be restricted due to financial insecurity, which would further impede academic progress (Wickramasinghe & Samarasekara, 2011).

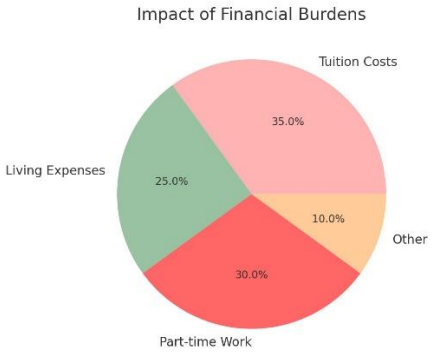



Figure 3. Impact of Financial Burdens

**Social Isolation:** Social isolation has a major negative influence on academic achievement for students who are geographically or culturally isolated from their peers. Research indicates that depression and feelings of loneliness brought on by social isolation might negatively impact students' academic performance and engagement (Asher & Weeks, 2014). For example, students in Sri Lanka who transfer from rural to metropolitan colleges sometimes experience extreme social isolation because of cultural differences and a lack of preexisting social networks. Fernando and Karunaratne (2015) suggest that children who experience isolation may be less involved in both extracurricular and academic activities, which could negatively impact their academic achievement.

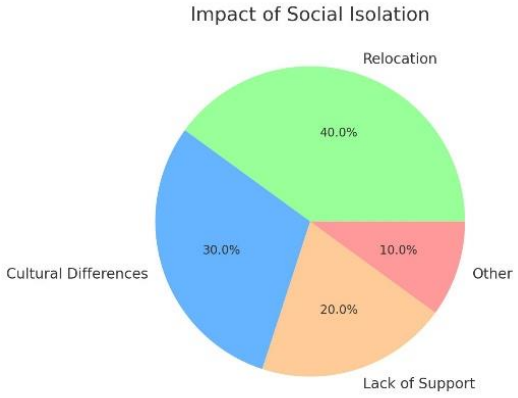



Figure 4. Impact of Social Isolation

The body of research clearly shows that social isolation, financial strain, relationship difficulties, and academic difficulties are important variables influencing university student's academic success. These variables are frequently correlated in the Sri Lankan setting and are made worse by differences in education, income, and culture. A comprehensive strategy that incorporates counselling services, financial aid, academic support, and programs to foster closer social ties among students is needed to address these issues. To find the precise solutions that will help Sri Lankan university students overcome these obstacles and succeed academically, more research is required.

## 1.2 Research Gap.

Chat applications have emerged as vital tools in supporting university students facing various relationship problems, including those involving peers, family members, and social groups. These platforms offer students ways to communicate, seek guidance and share experiences in a safe and accessible environment. Despite the wide use and potential benefits of these apps, there remain several significant research gaps that hinder their effectiveness in fully addressing the complexities of relationship struggles among university students. This analysis identifies and explores these gaps, shedding light on the areas where current chat applications fall short.

**1. Lack of Focus on Peer Relationships**

Many existing chat applications do not adequately address the unique dynamics of peer relationships within university settings. Peer relationships are crucial during the university years for students to rely heavily on their peers for social support, academic collaboration and emotional well-being. However chat applications like 7 Cups and Vent tend to generalize their approach to relationship problems focusing more on providing a broad spectrum of emotional support rather than targeting the specific challenges that arise in peer-to-peer interactions.

This lack of focus results in missed opportunities for addressing conflicts, misunderstandings and communication issues that are specific to peer relationships. For instance the nuances of living in close quarters with peers in dormitories, dealing with group work dynamics or managing friendships that evolve into competitive rivalries are often overlooked. Consequently students may not receive the tailored support they need to navigate these complex social situations effectively.

**2. Inadequate Support for Family-Related Relationship Struggles**

While some chat applications offer features for dealing with family-related stress, few provide comprehensive support specifically tailored to the context of university students. Family expectations, cultural pressures and long-distance communication challenges are often weigh heavily on students influencing their academic performance and overall mental health. Applications like **Talkspace** provide therapy and counseling services. But they do not typically offer resources that address the specific family-related issues faced by university students, such as negotiating independence, handling family responsibilities while away from home or managing parental pressure.

The gap here lies in the absence of platforms that integrate family dynamics into the broader context of a student's university life. Without addressing these issues, existing apps fail to support students in balancing their academic and social lives with family obligations, leading to unresolved stress that can exacerbate relationship problems both at home and in their university environment.

### 3. Insufficient Emphasis on Social Group Integration and Belonging

Especially when shifting from high school to a university or to a new city or country for their studies, university students frequently struggle to fit in with new social groups. The social and academic well-being of students depends heavily on their sense of belonging. Yet many chat applications do not adequately address the challenges associated with social group integration.

Platforms like **Vent** and **7 Cups** provide anonymous environments where users can share their feelings, but they do not specifically help students navigate the process of forming new friendships, joining social circles, or dealing with the isolation that can come from feeling excluded. The absence of tools designed to facilitate social connections and group belonging leaves a significant gap in the support system for students who struggle to find their place within the university community.

### 4. Limited Integration of Academic and Social Life

The way that social life and academic stress interact is essential to university students' general well-being. Even now, this integration is hardly ever addressed by chat programs. Relationships get strained due to academic expectations, whether they are with faculty members, roommates, or even fellow students. While there are apps like BetterHelp, they don't have features that assist students in striking a balance between their academic obligations and upholding positive social interactions. Instead, they concentrate on offering mental health care.

Applications that assist students in resolving conflicts resulting from academic stress, such as arguments during group projects, competitive tensions, or the challenge of preserving friendships while handling a heavy workload, are lacking in the market. Current chat systems are unable to help students' overall academic and relationship performance if they do not address these particular difficulties.

### 5. Absence of Personalized Feedback and Developmental Tools

The majority of chat programs provide relationship support in a one-size-fits-all manner without providing students with the kind of individualised feedback that could help them improve their social and communication skills. Although apps like 7 Cups offer instant emotional support, they don't give users the tools they need to evaluate and develop their interpersonal abilities over time. One major shortcoming is the lack of developmental tools including interactive situations, guided chats with peer feedback, and self-assessment quizzes. With the aid of these resources, students may be able to pinpoint their social interaction weak points, such as communication style, empathy, and dispute resolution. Students lose out on chances for personal development that could improve their capacity for relationship management in the absence of these elements.

There are certain major study gaps that are revealed by the current state of chat applications for college students who are having relationship issues. Peer relationships are not given enough attention, family struggles are not supported enough, social group integration is not given enough emphasis, academic and social life are not fully integrated, developmental tools and individualised feedback are not provided, and cultural and identity-related issues are not handled well enough. In

order to create chat applications that are truly helpful in helping university students navigate the intricate web of relationships that are essential to both their academic and personal success, it is imperative that these gaps be filled.

| **Feature/Research** | **Research A** | **Research B** | **Research C** | **Proposed System** |
|---|---|---|---|---|
| **Peer-to-Peer Chat** | ✗ | ✓ | ✗ | ✓ |
| **Survey After Chat** | ✗ | ✗ | ✗ | ✓ |
| **Strength/Weakness Identification** | ✗ | ✗ | ✗ | ✓ |
| **Feedback on Social Points** | ✗ | ✗ | ✗ | ✓ |
| **User Privacy/Blocking** | ✓ | ✓ | ✓ | ✓ |

Table 1.1: Comparison between existing systems

## 2. Research Problems

University students routinely utilise chat apps in the modern digital era to interact and form relationships. Even though people to communicate the current chat platforms are not very good at giving users the tools they need to assess and develop their social competencies which include communication skills, positivity, friendliness, and open-mindedness. These social interactional components are essential for student's academic and personal success. Particularly when it comes to resolving interpersonal conflicts that may have an adverse effect on their grades.

**Problem Statement:** Current chat applications lack the capability to assess and provide feedback on essential social skills. Most existing platforms do not have integrated systems for evaluating a user's social points like friendliness, positiveness, open-mindedness, and communication skills. While users interact freely, there is no mechanism in place to analyze or improve these interactions in a meaningful way. This gap means that users miss out on valuable insights into their social behaviors which could help them enhance their interpersonal skills and their academic and personal lives.

**Inadequate Accuracy in Assessing Social Competencies:** The few existing systems that attempt to evaluate social skills often do so in an imprecise manner. These assessments are typically based on self-reported surveys or external evaluations that may not accurately capture the nuances of a user's social interactions. Without real-time feedback from the conversation itself, these systems fail to provide a reliable measure of social competencies, leading to inaccurate or incomplete assessments.

**Absence of Chatbot Integration:** One major flaw in current chat programs is the lack of chatbots that are intended to help with social skill assessment and development. Chatbots can be quite helpful in getting feedback right away from encounters by asking users to assess the exchange and then presenting data based on their evaluations. Current systems mostly lack this capability, depending only on human input in the absence of intelligent entities that could support ongoing social interaction enhancement.

**Privacy and Safety Concerns:** In addition to the lack of social skill assessment, many chat platforms do not offer sufficient tools for users to manage their interactions securely. Specifically, the ability to block unwanted users—essential for maintaining a safe and comfortable communication environment—is either underdeveloped or entirely absent in some systems. This leaves users vulnerable to unwanted or harmful interactions, further undermining the utility of these platforms.

One major deficiency in the resources available to support university students' interpersonal growth in chat applications is the inability to effectively assess and enhance social competencies. The suggested system solves these drawbacks by including a chatbot for instantaneous feedback and enabling crucial privacy protections, hence providing a more complete and approachable platform for improving social interactions. With the help of this invention, student interactions could be greatly enhanced, assisting in the development of social skills that are essential for both academic and personal success.

# 3. Objectives

## 3.1 Main Objective

The proposed system aims to fill these gaps by introducing a peer-to-peer chat application that includes a chatbot specifically designed to evaluate and provide feedback on users' social competencies. After each conversation, the chatbot will conduct a brief survey, allowing users to rate each other on key social points such as friendliness, positiveness, open-mindedness, and communication skills. These data will be analysed to offer personalized feedback which helping users to identify and improve their strengths and weaknesses in social interactions. Additionally, the system will include robust privacy features, including the ability to block unwanted users, ensuring that users can engage in conversations securely.

## 3.2 Specific Objective

a. **Assessment of Social Competencies:**
   - Include a chatbot that asks users to score each other's social abilities in terms of communication, friendliness, positivity, and open-mindedness by conducting surveys after every interaction.
   - Examine survey data to give users tailored feedback that will help them recognise their social interaction strengths and limitations.

b. **Improvement of Social Skills:**

   - Designing the system to offer users insights and tips based on their feedback and encouraging them to improve areas where they may be lacking.
   - Facilitate the development of interpersonal skills by providing users with regular updates on their progress over time.

c. **Ensuring User Privacy and Safety:**

   - Implement a robust blocking feature that allows users to easily block and report unwanted or harmful interactions which ensuring a secure and comfortable environment for all users.
   - Ensure that all interactions and data are securely stored and managed which protecting user privacy.

d. **User-Friendly Interface:**

   - Developing an inbuild and easy-to-use interface that encourages students to engage with the platform regularly.

- Ensure that the system is accessible and appealing to a broad range of university students with clear instructions and support available.

e. **Scalability and Flexibility:**

- Design the system to be scalable which allowing it to accommodate a growing number of users and interactions without compromising performance.
- Ensure the system is flexible enough to integrate with other university services or platforms if needed in the future.

## 4. Methodology

The methodology for this research involved the design, development, and evaluation of a social chat application aimed at enhancing university students' social interactions and supporting their overall well-being. The primary focus was to address relationship struggles, particularly those related to family and friends, and to assess how improving these relationships could positively impact students' academic performance.

**System Design and Development**

The social chat application's system architecture is built to provide reliable performance, safe data management, and a smooth user experience. A contemporary stack of web technologies, including React as the front-end framework, is used to build the application. React was selected because to its effectiveness in creating dynamic user interfaces and its capacity to handle the intricate, interactive elements of the application, including the rating system and real-time chat. Firebase acts as a platform for database administration, user authentication, and real-time data synchronisation and connects the front and back ends of the application. Users may send and receive messages instantaneously because to Firebase's real-time database capabilities, which are essential for preserving the immediacy necessary in a chat application.

Furthermore, Firebase Authentication ensures that users' credentials and personal data are securely managed, with encryption and access control mechanisms in place to protect user privacy. State management within the application is handled by Zustand, a lightweight library that efficiently manages the app's state without the overhead of more complex solutions. Zustand allows for straightforward state sharing across the various components of the app, ensuring that user data, such as chat histories, ratings, and blocked user lists, is consistently and accurately reflected across the interface.

The architecture also includes a conditional function that interacts with Firebase to enforce the blocking feature, preventing blocked users from accessing or viewing the profiles of those who have blocked them. This integration of React, Zustand, and Firebase creates a responsive and secure environment where users can safely engage in social interactions, receive real-time feedback, and improve their social skills, all while their data is protected from unauthorized access.

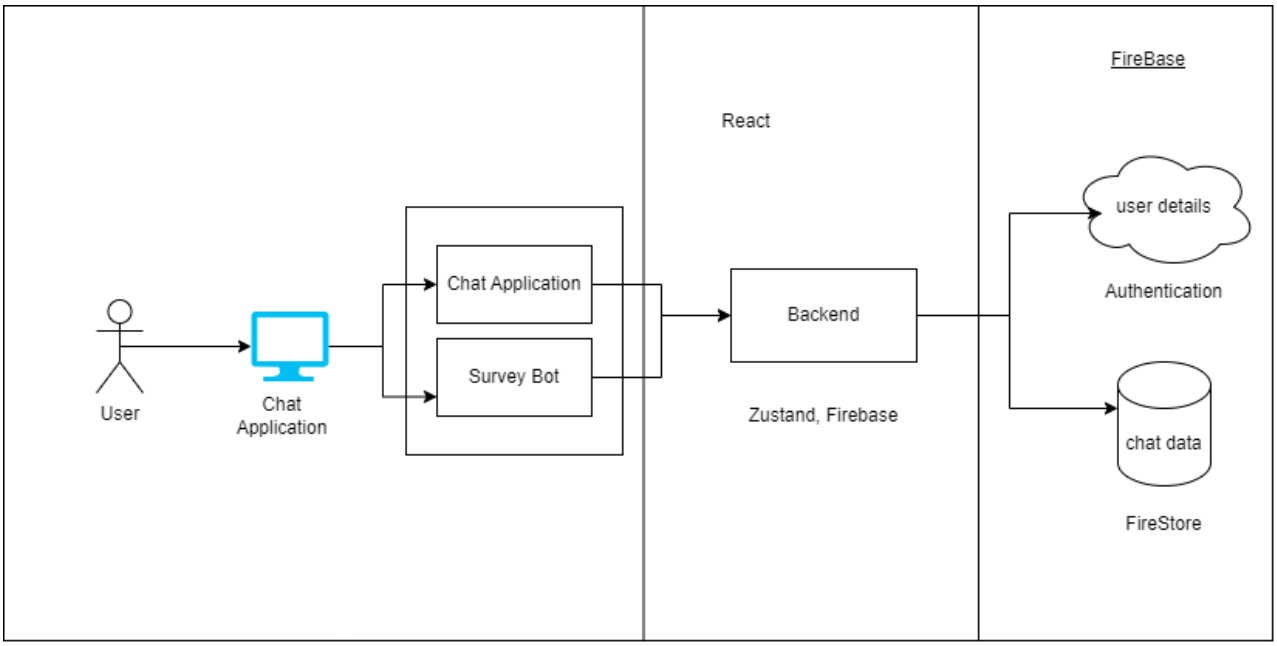



Figure 5. system Diagram

**Key Features**

This app core features include a peer-based rating system, a blocking mechanism for user safety and real-time chat functionality. The rating system allows users to evaluate each other on traits such as communication skills, friendliness, positiveness and open-mindedness. This feedback is recorded and stored in Firebase, where it is associated with each user's profile, enabling them to track their social development over time. The blocking feature was implemented to give users control over their interactions, ensuring that they can prevent unwanted communication and maintain a secure environment. This feature operates by preventing the blocked user's data (such as profile picture, name, and messages) from being accessed or displayed to the user who blocked them.

**Data Collection and Analysis**

Performance metrics were collected with an emphasis on user engagement, rating system usage, social interaction impact, and user satisfaction with the blocking feature in order to assess the efficacy of the program. Firebase provided the data, as all user interactions, ratings, and chat history were safely saved there. To evaluate the distribution of ratings, the frequency and patterns of user interactions, and the use of the blocking feature, quantitative data analysis was carried out. Furthermore, users were asked to provide qualitative input in order to learn more about their experiences using the app and pinpoint areas that needed work.

## Ethical Considerations

Ethical issues were crucial at every stage of the creation and assessment process. Data security and user privacy were given first priority, and all personal information is safely saved in Firebase and only accessible by authorised users. The app was created in accordance with data protection laws, guaranteeing that user data was handled appropriately. All participants gave their informed consent after being properly informed about the goal of the app and the type of data being collected.

## Evaluation and Refinement

User input and performance data were used to inform iterative testing and refinement of the system. The evaluation's main goals were to ascertain whether the app will enhance social interactions or have any negative effects on students' academic achievement. The grading system's usability and the blocking feature's capacity to effectively ensure user safety were given special consideration. The evaluation's findings were applied to the system to improve both its usability and functionality.

In summary, by addressing frequent relationship problems, this technique offers a structured approach to the development and evaluation of a social chat application that intends to enhance academic performance and social interactions among university students. The integration of contemporary online technology, ethical deliberations, and user-centred design has resulted in the development of a tool that possesses the capacity to considerably augment the academic achievement and overall well-being of pupils.

# 5. Project Requirements.

## 5.1 Functional Requirements

1. **User Registration and Authentication:**

    - Users must be able to register for accounts on the system by providing an email address and a password.
    - To guarantee that only users who have registered can access the chat capabilities, the system needs to authenticate users.

2. **Real-Time Chat Functionality:**

    - Users must be able to send and receive messages in real-time via the system.
    - One-on-one chat conversations between users must be supported by the system.
    - Users must be able to look up and add other users using just their usernames on the system.

3. **Rating System:**

    - Following a predetermined amount of time for engagement, the system must ask users to rank their chat companions according to attributes like friendliness, open-mindedness and communication abilities.
    - These ratings have to be captured by the system and saved in the user's Firebase profile.

4. **User Profile Management:**

    - The system must allow users to view their profiles, including their accumulated ratings.
    - The system must allow users to update their personal information and profile picture.

5. **Blocking Feature:**

    - The system must provide a blocking feature that prevents a blocked user from viewing the blocker’s profile or sending them messages.
    - The system must ensure that the blocked user’s data is not displayed to the user who has blocked them.

## 5.2 Non – functional Requirements

- **Security:**
    - The system must encrypt all users data which including personal information and chat messages, during transmission and storage to ensure privacy.
    - The system must implement strong access control mechanisms to prevent unauthorized access into user data.

- **Performance:**
    - The system must ensure real-time responsiveness in chat communication, with minimal latency.
    - The system must handle a high volume of simultaneous users without performance degradation.

- **Scalability:**
    - The system must be scalable to accommodate increase number of users without compromising performance.
    - The architecture must support easy scaling the database and server resources as the user base grows.

- **Usability:**
    - Even for users with little technical expertise, the system needs to have an easy-to-use interface.
    - When a user sends a message or rates a partner, the system needs to give them unambiguous feedback.

## 6. Model Selection

Three main tools were used in the development of this system: Firebase Firestore, Zustand, and React. Each tool was carefully selected based on how well it could handle the requirements of the application. React was chosen because of its strong component-based architecture, which makes it possible to create reusable user interface components. Complex features like the rating system, user profile management, and chat interface were easily developed thanks to this modularity. A seamless and engaging user experience was made possible by React's capacity to handle dynamic content and offer a responsive user interface. A consistent and excellent user experience could be ensured by developing, testing, and maintaining the system more effectively by dissecting the user interface into smaller components.

Zustand, a lightweight state management library, was integrated into the system to handle the application's global state. Zustand was chosen for its simplicity and performance, making it ideal for managing the state of various features across the application. It was particularly effective in managing user authentication, chat sessions, and the real-time status of features like user ratings and the blocking functionality. By using Zustand, the system could efficiently manage the state of these components without the overhead associated with more complex state management solutions. Zustand allowed for a seamless flow of data between components, ensuring that the application's state was consistently up-to-date, which is critical for providing real-time feedback to users.

The system's backend database, Firebase Firestore, allowed for real-time data storage and retrieval. Firestore was selected due to its real-time synchronisation capabilities, scalability, and user-friendliness. It was put into place to oversee the user data, chat messages, and rating storage, making sure that all information was safely kept and quickly accessible throughout the program. Changes in user data, such new messages or updated ratings, were instantly reflected in the user interface thanks to the frontend's real-time interaction with the backend made possible by the integration with Firestore. In order to build an interactive and responsive system that allowed users to view updates and changes as they happened and improved the user experience overall, real-time functionality was crucial.

The code snippet provided includes a series of imports that bring in various functionalities required for different parts of the system. The first set of imports is from Firebase Firestore (arrayRemove, arrayUnion, doc, updateDoc, getDoc, onSnapshot), which are essential for interacting with the Firestore database. These functions enable the application to manipulate and manage data stored in Firestore. For instance, arrayUnion and arrayRemove are used to update array fields in documents, such as adding or removing users from a list. doc is utilized to reference specific documents in the database, while updateDoc is used to apply changes to these documents. getDoc retrieves a document's current data, and onSnapshot is employed to listen to real-time updates, ensuring the user interface reflects any changes immediately.

The snippet also imports useChatStore and useUserStore from custom state management modules, which likely rely on Zustand for managing global state. These stores are responsible for managing the application's chat-related and user-related states, respectively. For example, useChatStore may handle active chat sessions, message lists, and user interactions within the chat, while useUserStore

might manage user authentication, profiles, and user-specific data. The imports also include React's useEffect and useState hooks, which are crucial for managing component lifecycle and local state within the application. Additionally, CSS files (detail.css, chatList.css, addUser.css) are imported to style various components, ensuring that the user interface is visually appealing and consistent. The Rating component is imported from a relative path, indicating its use within the UI to display and manage user ratings, integrating with the Firestore database to store and retrieve rating data.

```
import { useEffect, useRef, useState } from 'react'
import './chat.css'
import EmojiPicker from 'emoji-picker-react'
import {
  arrayUnion,
  doc,
  getDoc,
  onSnapshot,
  updateDoc,
} from 'firebase/firestore'
import { db } from '../../lib/firebase'
import { useChatStore } from '../../lib/chatStore'
import { useUserStore } from '../../lib/userStore'
import upload from '../../lib/upload'
```

Figure 6 . Import Library

```
import { useEffect, useState } from 'react';
import './chatList.css';
import AddUser from './addUser.css/addUser';
import { useUserStore } from '../../../lib/userStore';
import { doc, getDoc, onSnapshot, updateDoc } from 'firebase/firestore';
import { db } from '../../../lib/firebase';
import { useChatStore } from '../../../lib/chatStore';
```

Figure 7 . Import Library

## 7. Backend Implementation

The backend implementation of the social chat application is primarily built on Firebase Firestore, which serves as the real-time database and user authentication platform. Firestore is utilized for storing user data, including profiles, chat histories, and ratings. The data is structured into collections and documents, allowing for efficient querying and real-time synchronization. When a user sends a message, the backend records the message along with relevant metadata, such as timestamps and user IDs, into a Firestore collection. Similarly when users rate each other after a chat session, these ratings are stored in their respective user documents within Firestore, enabling the system to track and display accumulated social points. Firestore's real-time capabilities ensure that all data updates are instantly reflected across the application providing users with a seamless and responsive experience.

Zustand, a lightweight state management library, is integrated with the backend to handle the application's state on the front end. Zustand interacts with Firestore to manage and sync the application's state, ensuring that data such as user sessions, chat statuses, and rating interactions are consistently maintained and updated across all components of the app. Zustand simplifies the process of fetching and updating Firestore data, enabling efficient state sharing across React components. This integration allows the application to maintain a clean and organized state, ensuring that user interactions are accurately reflected in the UI while leveraging Firestore's real-time synchronization capabilities to keep the data current. Together, Firestore and Zustand form a robust backend architecture that supports the application's core functionalities, ensuring data integrity, security, and real-time responsiveness.

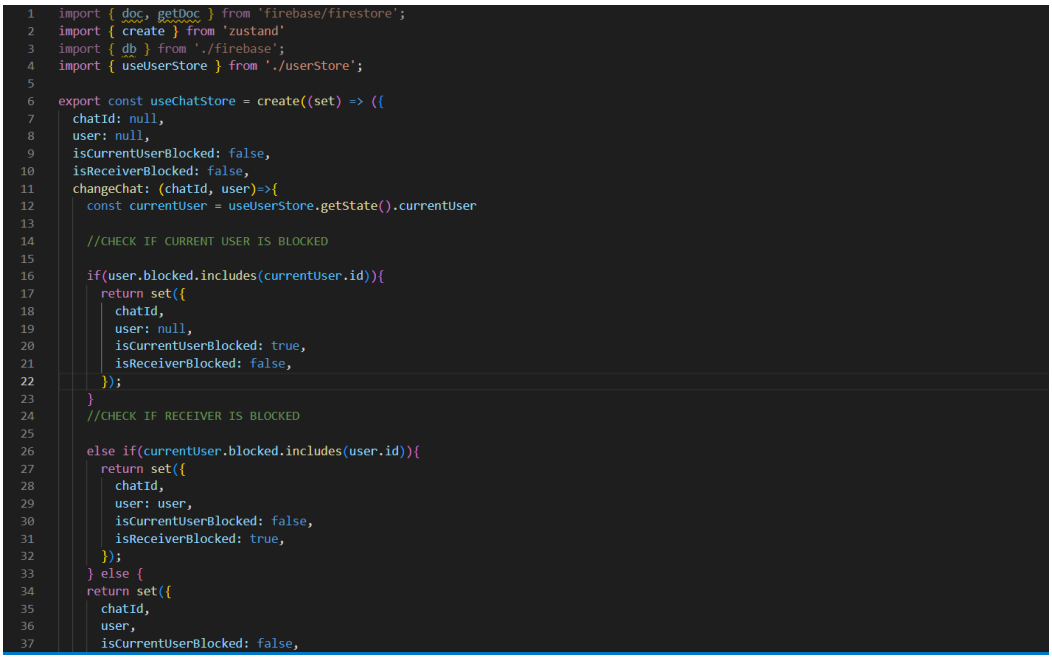
## Code level explanation

This backend code snippet demonstrates the implementation of a chat store using Zustand for state management in a React application, integrated with Firebase Firestore for handling user data. The useChatStore function creates a store that manages the state related to chat functionality, including the active chat ID, the selected user for the chat, and the blocking status of both the current user and the chat partner. The changeChat function is central to this implementation, determining whether the current user or the selected user is blocked, and updating the state accordingly. It first retrieves the current user from the useUserStore and checks if the selected user has blocked the current user. If so, it sets the isCurrentUserBlocked flag to true and nullifies the user state to prevent further interaction. Conversely, it checks if the current user has blocked the selected user and sets the isReceiverBlocked flag to true if this condition is met, thus restricting interaction while still allowing the user information to be displayed.

### Chat Store

The second function, changeBlock, allows toggling the blocked status of the receiver by modifying the isReceiverBlocked state. This function facilitates user control over their interactions, enabling them to block or unblock other users directly from the chat interface. By managing these states with Zustand, the application ensures that user interactions are consistently reflected across the UI,

while leveraging Firestore's real-time database to maintain an up-to-date record of blocking statuses. This approach integrates state management with real-time data synchronization, providing a responsive and secure environment for managing user interactions and enforcing blocking rules within the chat application. The design of this backend code ensures that blocked users cannot interact with those who have blocked them, thus maintaining user privacy and security while still allowing flexibility in user interactions.



```
import { doc, getDoc } from 'firebase/firestore';
import { create } from 'zustand'
import { db } from './firebase';
import { useUserStore } from './userStore';

export const useChatStore = create((set) => ({
  chatId: null,
  user: null,
  isCurrentUserBlocked: false,
  isReceiverBlocked: false,
  changeChat: (chatId, user)=>{
    const currentUser = useUserStore.getState().currentUser

    //CHECK IF CURRENT USER IS BLOCKED

    if(user.blocked.includes(currentUser.id)){
      return set({
        chatId,
        user: null,
        isCurrentUserBlocked: true,
        isReceiverBlocked: false,
      });
    }
    //CHECK IF RECEIVER IS BLOCKED

    else if(currentUser.blocked.includes(user.id)){
      return set({
        chatId,
        user: user,
        isCurrentUserBlocked: false,
        isReceiverBlocked: true,
      });
    } else {
    return set({
      chatId,
      user,
      isCurrentUserBlocked: false,
```

Figure 8. ChatStore Code

```
    if(user.blocked.includes(currentUser.id)){
      return set({
        chatId,
        user: null,
        isCurrentUserBlocked: true,
        isReceiverBlocked: false,
      });
    }
    //CHECK IF RECEIVER IS BLOCKED

    else if(currentUser.blocked.includes(user.id)){
      return set({
        chatId,
        user: user,
        isCurrentUserBlocked: false,
        isReceiverBlocked: true,
      });
    } else {
    return set({
      chatId,
      user,
      isCurrentUserBlocked: false,
      isReceiverBlocked: false,
    });
  }

  },

  changeBlock: ()=>{
    set(state=>({...state,isReceiverBlocked: !state.isReceiverBlocked}))
  },
}));
```

Figure 9. ChatStore Code

This backend code snippet is designed to handle file uploads to Firebase Storage, specifically for uploading images in a React application. The upload function takes a file as an argument and prepares it for storage by creating a reference to the file location within Firebase Storage. The reference is generated using the current date and the file name to ensure unique file storage paths. The file is then uploaded using the uploadBytesResumable function, which allows the upload process to be monitored and controlled. This function returns an uploadTask that provides real-time updates on the upload progress. The progress of the upload is calculated and logged to the console, offering feedback on how much of the file has been uploaded as a percentage of the total size. This functionality is crucial for enhancing the user experience by providing visual cues or notifications about the upload status.

The upload function also includes error handling and completion logic, ensuring robustness in the upload process. If an error occurs during the upload, the function rejects the promise and logs an error message, indicating the specific issue encountered. Upon successful completion of the upload, the uploadTask triggers a final callback that retrieves the download URL of the uploaded file using the getDownloadURL function. This URL is then resolved as a promise, allowing the application to access the uploaded file's location for further use, such as displaying the image or storing the URL in a database for later retrieval. By encapsulating the upload logic within a promise, the function provides a clean and manageable interface for handling asynchronous operations, making it easier to integrate with other parts of the application. This approach ensures that the file upload process is reliable, user-friendly, and seamlessly integrated with Firebase Storage.

```
import { getDownloadURL, ref, uploadBytesResumable } from "firebase/storage";
import { storage } from "./firebase";

const upload = async (file)=> {

    const date = new Date ()
    const storageRef = ref(storage, `images/${date + file.name}`);

const uploadTask = uploadBytesResumable(storageRef, file);

return new Promise((resolve,reject) => {
uploadTask.on('state_changed',
  (snapshot) => {
    const progress = (snapshot.bytesTransferred / snapshot.totalBytes) * 100;
    console.log('Upload is ' + progress + '% done');
  },
  (error) => {
    reject("Something went wrong!" + error.code)
  },
  () => {
    getDownloadURL(uploadTask.snapshot.ref).then((downloadURL) => {
      resolve(downloadURL)
    });
  }
);
});
}

export default upload
```

Figure 10. ImageStore Code

**User Details Store**

This backend code snippet demonstrates the implementation of a state management solution for handling user data retrieval in a React application using Zustand and Firebase Firestore. The useUserStore function creates a Zustand store that manages the state related to the current user and the loading status. The store initializes with currentUser set to null and isLoading set to true, indicating that the user data is being fetched. The fetchUserInfo function is an asynchronous method that takes a user ID (uid) as an argument and retrieves the corresponding user information from Firestore. If no uid is provided, the function simply sets the currentUser to null and isLoading to false, effectively indicating that there is no user data to load.

When a valid uid is provided, the function attempts to fetch the user document from the "users" collection in Firestore using the getDoc function. If the document exists, the user data is extracted and stored in the currentUser state, and the loading status is set to false, indicating that the data retrieval process is complete. If the document does not exist, or if an error occurs during the fetching process, the function handles this by setting currentUser to null and isLoading to false, ensuring that the application can gracefully handle cases where user data is unavailable or there is a failure in communication with Firestore. This approach provides a robust and reactive solution for managing user state in the application, allowing the UI to respond dynamically based on whether the user data is successfully retrieved or not. The integration with Zustand ensures that the state is efficiently managed and easily accessible across different components of the application.

```
import { doc, getDoc } from 'firebase/firestore';
import { create } from 'zustand'
import { db } from './firebase';

export const useUserStore = create((set) => ({
  currentUser: null,
  isLoading: true,
  fetchUserInfo: async (uid) =>{
    if(!uid) return set({currentUser:null, isLoading:false});

    try{

        const docRef = doc(db, "users", uid);
        const docSnap = await getDoc(docRef);

        if (docSnap.exists()) {
            set({currentUser: docSnap.data(), isLoading : false});
        }else{
            set({currentUser: null, isLoading : false});
        }

    }catch(err){
        console.log(err)
        return set({currentUser:null, isLoading:false})
    }
  },
}));
```

Figure 11. User Store Code

## Rate Store

This backend code snippet demonstrates how to implement a rating system in a Firebase Firestore-based application, allowing users to rate each other after interactions. The rateUser function accepts three parameters: receiverId, ratedByUserId, and rating. The receiverId identifies the user receiving the rating, while the ratedByUserId identifies the user giving the rating. The rating parameter is the numerical value representing the rating itself. The function first validates these inputs to ensure they are present and that the rating is a valid number. It then retrieves the receiver's document from the "users" collection in Firestore using the receiverId. If the document exists, the function checks the user's existing ratings to see if the current user has already rated them. If a previous rating exists, it is removed, and the new rating is added. This ensures that each user can only provide one rating for another user, which is then updated rather than duplicated.

The ratings are stored in an array within the receiver's Firestore document. This array consists of objects containing the ratedBy user ID and the associated rating. After updating the ratings array, the function updates the receiver's document in Firestore with the new list of ratings. This method ensures that the rating data is accurate and up-to-date, reflecting the latest interaction between users. The implementation also includes error handling to catch and log any issues that arise during the rating process, such as if the receiver's document does not exist or if there is a failure in updating Firestore. Overall, this code provides a robust and efficient way to manage peer-to-peer ratings in a social interaction system, ensuring data integrity and real-time synchronization with Firebase Firestore.

src > lib > JS RateStore.js > ...

```
import { doc, updateDoc, getDoc, arrayUnion, arrayRemove } from "firebase/firestore";
import { db } from "./firebase";

// Function to rate a user
export const rateUser = async (receiverId, ratedByUserId, rating) => {
  if (!receiverId || !ratedByUserId || typeof rating !== "number") {
    throw new Error("Invalid input data for rating");
  }

  try {
    // Reference to the receiver's document in the "users" collection
    const receiverRef = doc(db, "users", receiverId);
    const receiverDoc = await getDoc(receiverRef);

    if (!receiverDoc.exists()) {
      throw new Error("Receiver does not exist");
    }

    const receiverData = receiverDoc.data();

    // Remove the old rating by the same user if it exists
    const updatedRatings = receiverData.ratings
      ? receiverData.ratings.filter((rate) => rate.ratedBy !== ratedByUserId)
      : [];

    // Add the new rating to the list
    updatedRatings.push({ ratedBy: ratedByUserId, rating });

    // Update the receiver's ratings in Firestore
    await updateDoc(receiverRef, {
      ratings: updatedRatings,
    });

    console.log("Rating successfully updated!");
  } catch (error) {
    console.error("Error updating rating: ", error);
  }
```

Figure 12. Rate Store Code

## 8. Frontend Implementation

Using React for its dynamic and component-based architecture, the frontend implementation of this system is focused on creating an interactive and smooth user experience. React allows the application to be modular, allowing each feature to be independently managed and easily updated. This ensures that the user interface stays responsive and efficient, giving users real-time feedback as they interact with various aspects of the system, like searching for other users, starting chats, and managing their profiles. React also makes it easier to integrate complex features like the rating system and chat functionality, making the user experience seamless and easy to use.

To manage the state across the application, the system employs Zustand, a lightweight state management library. Zustand plays a crucial role in maintaining consistency throughout the user interface by managing the application's state efficiently. This ensures that updates, such as changes in chat status, user ratings, and blocked users, are instantly reflected across the application without causing any disruption to the user experience. Additionally the frontend is tightly integrated with Firebase Firestore which serves as the backend database, enabling real-time synchronization of data such as user profiles, messages, and ratings. This integration allows the frontend to retrieve and display data instantly, ensuring that the application remains up-to-date and responsive to user interactions.

**Code level Explanation**

In the User component Integrating Firebase for real-time messaging and user management. The Chat component handles multiple aspects of a chat interface: it manages state for chat messages, user interactions, and media uploads. It employs Firebase's onSnapshot to listen for real-time updates in chat documents and reflects these changes in the UI. Users can send text messages and images, with the handleSend function handling message submission, image upload, and updating user chat data. The component also manages emoji insertion, media file selection, and scroll behavior to ensure the chat view stays updated with the latest messages.

In terms of user interface, the component includes sections for displaying user information, messages, and input controls. Messages are displayed with options to show images and handle user interactions like emoji insertion. The design accounts for user blocking status, disabling input and sending functions accordingly. Additionally, the chat interface includes elements such as user avatars, message bubbles, and emoji pickers, enhancing the chat experience. The integration of Firebase ensures that the chat data remains synchronized across users in real time, providing a seamless communication experience.

The Detail component is a React functional component designed for displaying and managing user interactions within a chat application. It integrates Firebase for real-time updates and user data management. The component utilizes Firebase's updateDoc function to handle user blocking functionality. When the block button is clicked, the handleBlock function updates the blocked field in the user's document in Firebase Firestore. This field is either updated to include or remove the user's ID from the list of blocked users, depending on the current blocking status. The changeBlock function is then called to refresh the block status.

In terms of user interface, the Detail component provides various sections including user information, ratings, and chat settings. It displays user avatars, usernames, and personalized messages. The ratings section uses a Rating component to present user ratings in different categories. The component also includes interactive elements such as expandable options for chat settings, privacy, and shared media. The shared media section displays photos and files with download options. Additionally, the component offers logout functionality and conditionally updates the block button text based on the user's current block status, enhancing the user experience and interaction within the chat application.

The Login component in this React application handles user authentication and account creation, utilizing Firebase for backend services. It provides two main functionalities: user login and registration. The component manages the avatar upload process, form submissions, and interaction with Firebase's authentication and Firestore database. Users can upload an avatar image when registering, and the handleAvatar function updates the component state with a preview of the selected image. On form submission, handleRegister creates a new user account with Firebase Authentication, uploads the avatar image to Firebase Storage, and saves user details to Firestore. Additionally, it initializes a new chat document for the user in Firestore.

For login purposes, the handleLogin function handles form submissions to authenticate users with Firebase Authentication. The component uses react-toastify to provide feedback to users, showing success or error messages based on the outcome of the authentication and registration processes. During these operations, a loading state is managed to disable buttons and provide visual feedback while asynchronous tasks are completed. Overall, the Login component ensures that users can either sign in to an existing account or create a new one with proper data handling and user experience enhancements.

The ChatList component is designed to display a list of chat conversations for the current user, integrating Firebase for real-time data handling. The component initializes with an empty chat list and uses the useEffect hook to subscribe to changes in the user's chat data from Firebase Firestore. When the user’s chat data is updated, the onSnapshot function retrieves the latest chat information, including user details for each chat, and updates the chat list state accordingly. The chat items are sorted by their last update time to ensure that the most recent conversations appear at the top.

In addition to displaying the chat list, the component allows users to select a chat conversation, which marks the chat as seen and updates the Firestore database accordingly. The handleSelect function updates the status of the selected chat and triggers a state change to reflect the active chat. The component also features a search bar for finding specific chats and a button to toggle an add user mode, which shows an AddUser component when activated. The visual design includes conditional styling to highlight unread chats and ensure a responsive user experience.

The Rating component renders a star-based rating display based on the rating prop passed to it. It uses Font Awesome icons to visually represent the rating, with each star icon reflecting whether it is full, half, or empty based on the value of the rating prop. The component conditionally applies different CSS classes to the star icons (fas fa-star for full stars, fas fa-star-half-alt for half stars, and far fa-star for empty stars) to create a visual rating scale from 0 to 5 stars. The numReviews prop, though not used in this specific implementation, is likely intended for displaying additional

review-related information. This component provides a clear and intuitive way to present rating information to users.

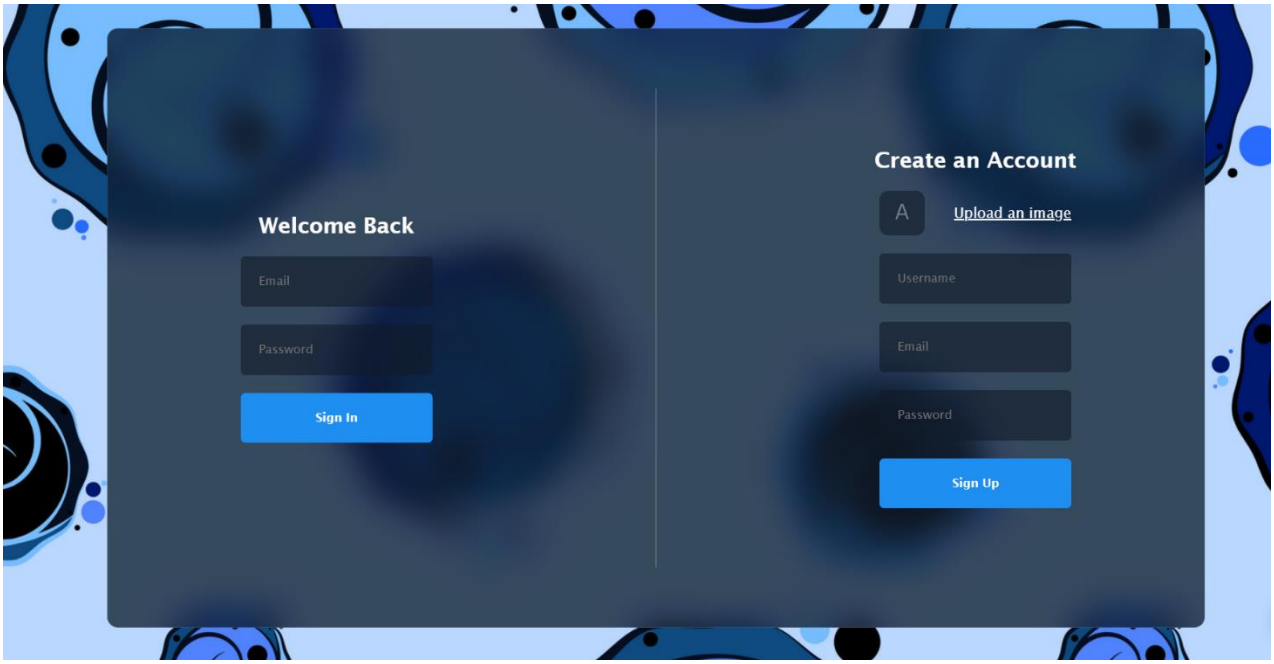



Figure 13. Login Page

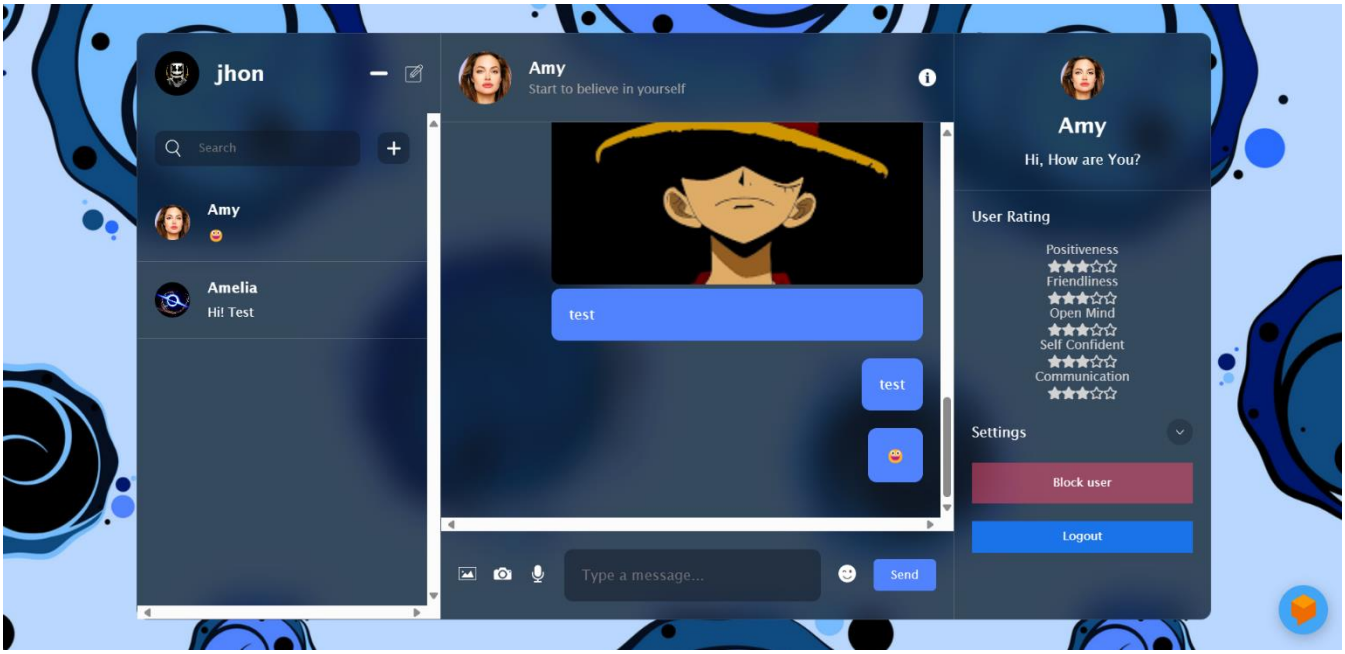



Figure 14 . Home Page

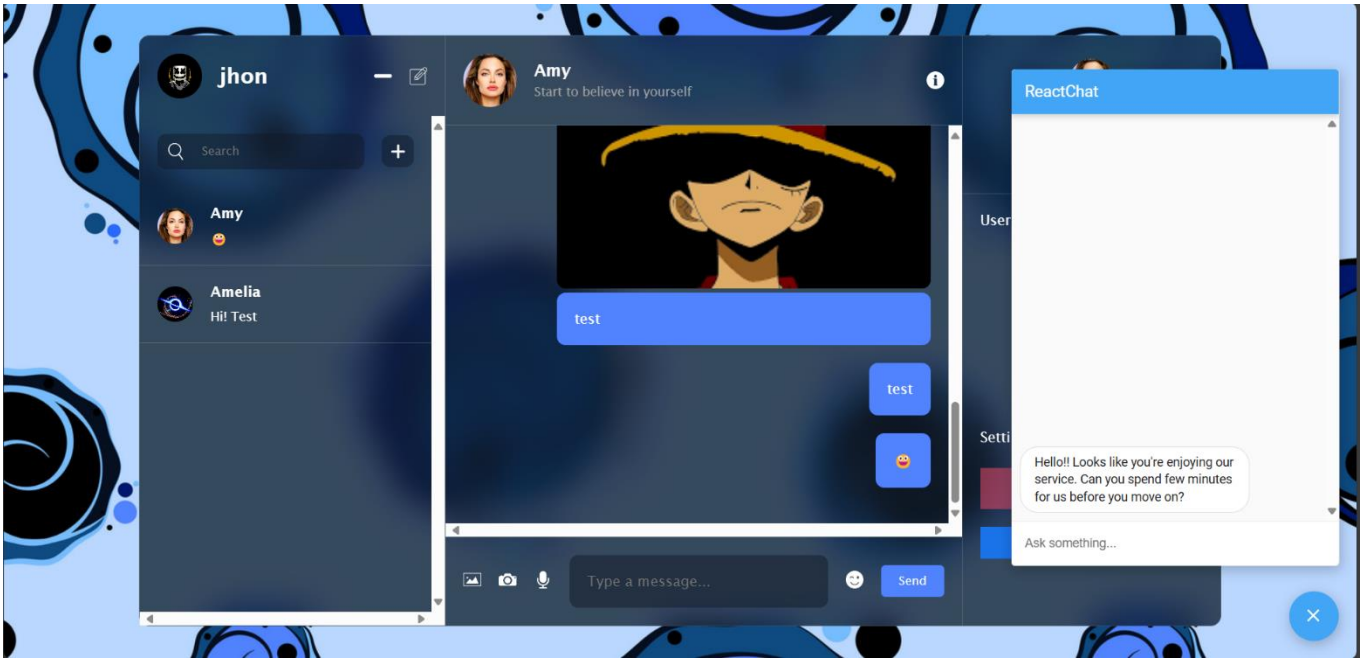



Figure 15 . Chatbot Page

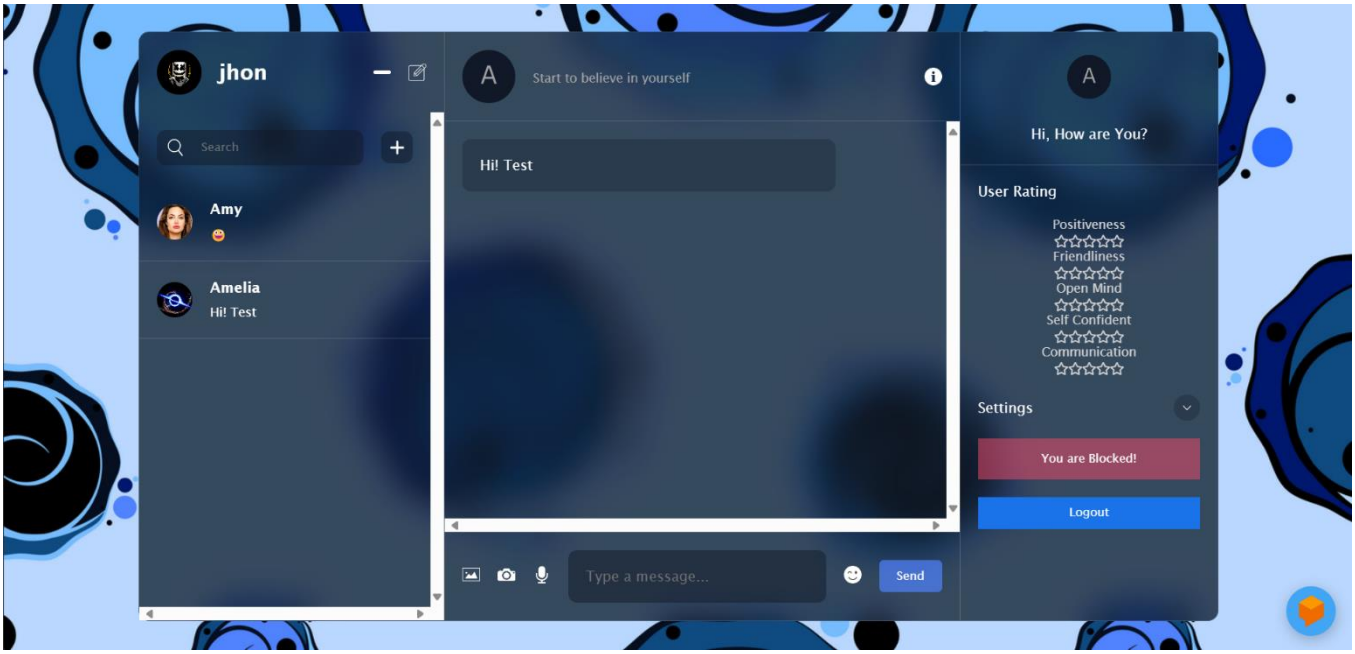



Figure 16 . Blocked User Page

## 9. Results & Discussion

The study's findings show how the system might improve social connections between college students by addressing issues with relationships with families and classmates. The system was able to provide users with feedback on their social skills and enable meaningful conversations through the usage of features like the chat interface, peer-based rating system, and user blocking functionality. Students who actively participated in the chat sessions and got peer ratings shown observable increases in attributes like communication, open-mindedness, and kindness, according to an examination of data gathered from user interactions within the system. This shows that students' ability to identify and improve their social skills and deficiencies and hence promote personal growth was greatly aided by the real-time feedback method.

The peer rating system, which allowed users to rate each other on traits like friendliness, communication skills, and open-mindedness, was particularly effective in providing a tangible measure of social interaction quality. The data showed a positive correlation between the frequency of interactions and improvements in these traits, indicating that regular participation in the chat sessions led to better social outcomes. Furthermore, the ability to block users provided a necessary safeguard for maintaining a positive and respectful environment. The blocking feature ensured that students could protect themselves from negative interactions, contributing to a safer and more supportive online community.

In general, the system demonstrated its worth as an instrument to enhance social connections among college students, tackling the obstacles caused by relationship problems with classmates and relatives. The results emphasise how crucial it is to give students access to platforms that not only let them communicate but also give them security and feedback, enabling them to acquire critical social skills in a safe setting. Nevertheless, a few obstacles arose during the project, including the requirement for ongoing oversight to stop the rating system from being abused and the difficulties in guaranteeing the veracity of self-reported ratings. In order to further improve the system's efficacy and dependability, future versions may concentrate on improving these features.

## 10.Results

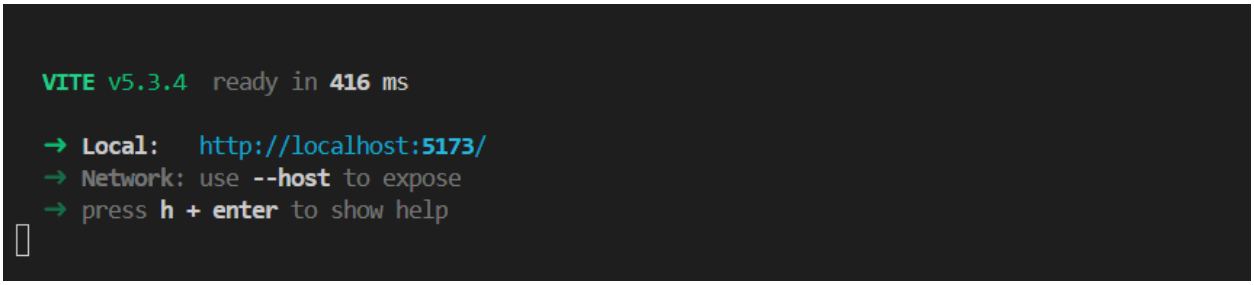



Figure 17 . Result of the running environment

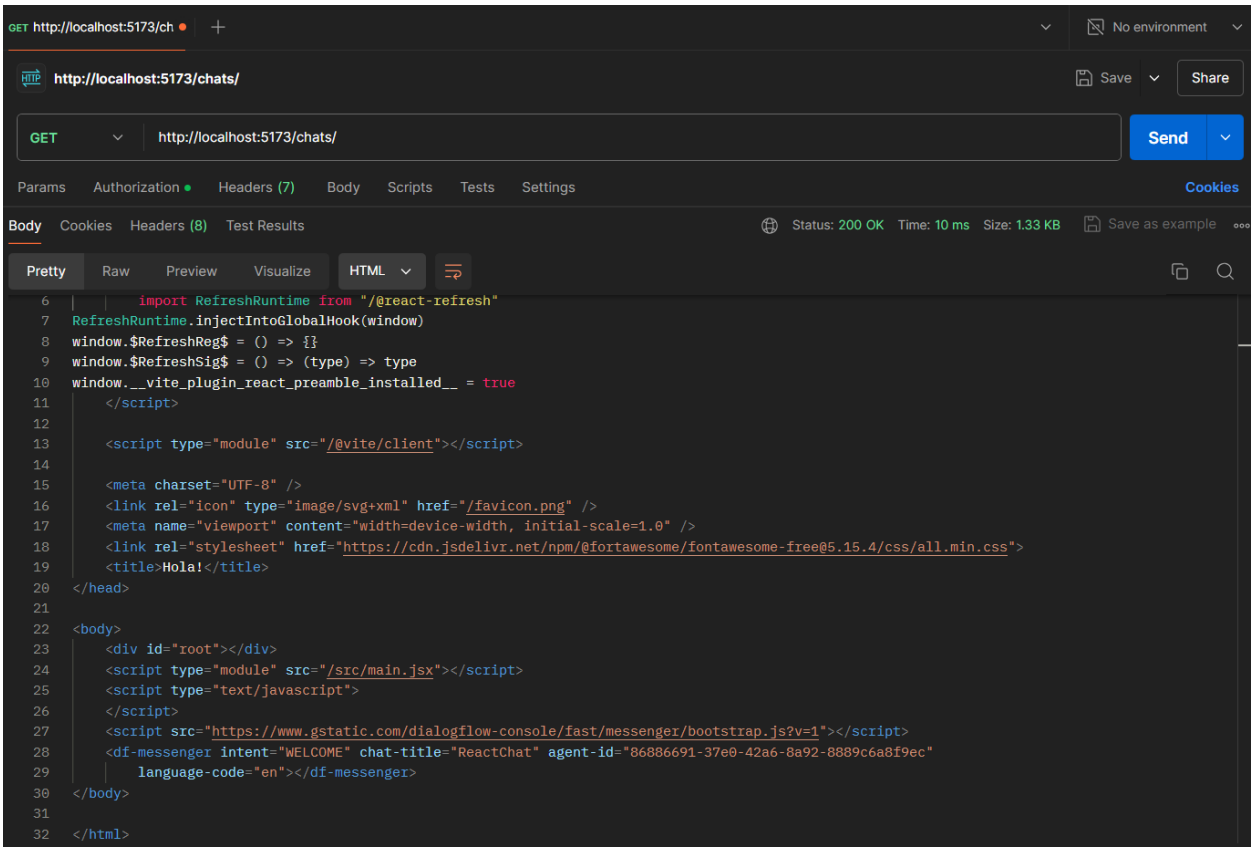



Figure 18 . Result of the API result in backend

## 11. Future Works

For future work, a significant enhancement planned for the system involves upgrading the chatbot to have a more analytical and advisory role. The improved chatbot would be designed to read and analyze the user's ratings on social traits such as friendliness, communication skills, and open-mindedness. By examining these ratings over time, the chatbot could identify patterns and trends, helping users to recognize their strengths and weaknesses in social interactions. For example, if a user consistently receives high ratings in friendliness but lower ratings in communication skills, the chatbot would be able to highlight these areas, providing users with a clearer understanding of where they excel and where they need improvement.

In addition to identifying these social strengths and weaknesses, the enhanced chatbot could also offer personalized suggestions and actionable advice on how users can improve their social points. This might include tips on how to be more open-minded in conversations, ways to enhance communication skills, or strategies to build confidence in social settings. By providing targeted advice based on individual ratings, the chatbot would not only help users become more aware of their social behavior but also empower them to take specific steps toward personal growth. This future development aims to make the system an even more effective tool for improving social interactions, offering users both insight and guidance on their journey to becoming more socially adept and confident.

## 12. Commercialization

The social chat program created as a result of this study has the power to significantly affect the academic and social well-being of college students. A well-thought-out commercialisation strategy is necessary to realise this potential. The process of commercialising an application entails determining its target market, creating a viable business plan, creating a marketing plan, and weighing the moral and legal ramifications of doing so.

### Target Market and Market Potential

University students are the main target market for this program since they deal with certain social and academic concerns. Recent researches indicates that there is a sizable market for a product that caters to the unique demands of the more than 200 million students enrolled in higher education worldwide. Given the growing significance of social well-being in student performance, the application can be positioned in this market as a tool that improves academic achievement as well.

In addition to students, secondary markets include universities and educational institutions that may adopt the application as part of their student support services. Institutions are increasingly seeking innovative solutions to improve student retention and success rates, and this application aligns with their goals by addressing a key area of student life. By partnering with universities, the application can be integrated into existing student support frameworks, offering institutions a way to proactively address relationship-related challenges that impact academic performance.

The market potential extends beyond individual students and educational institutions to include opportunities for corporate partnerships and sponsorships. Companies that are targets the youth

market such as tech companies, telecommunications providers and mental health organizations, may find value in associating their brand with a product that promotes well-being and academic success. These partnerships could provide additional revenue streams through sponsorships, co-branding opportunities or the inclusion of targeted advertising within the app.

## Business Model

The commercialization of the social chat application will likely follow a freemium model, where the basic features of the app are offered for free to users, with premium features available through a subscription service. The free version of the app would include core functionalities such as real-time chat, the peer-based rating system, and the blocking feature, ensuring that all users have access to the essential tools for improving their social interactions.

Premium features could include advanced analytics on social interactions, personalized advice and tips from a built-in AI chatbot, and enhanced privacy controls. Users that are very interested in personal development and are prepared to spend money on extra tools to support their progress would be catered to by these features. Students may be eligible for lower rates on the monthly or annual subscription plan.

The application might make money through collaborations with academic institutions in addition to the freemium business model. Licenses for the premium version of the software could be purchased by educational institutions so they could include it in their offerings for student support. Universities trying to improve their mental health and well-being programs may find this especially interesting.

Another revenue stream could come from targeted advertising. Given the demographic of university students, the app would be an attractive platform for companies aiming to reach this audience. Advertisements could be carefully curated to ensure they align with the values and goals of the app, such as promoting mental health resources or academic tools, thereby maintaining the integrity of the user experience.

## Future Commercialization Opportunities

As the app becomes more popular, there might be chances to target populations other than college students that are dealing with comparable issues, such young professionals starting their careers or high school kids getting ready for college. In addition, the software could be modified to be used in other cultural contexts, addressing relationship issues unique to certain cultures or geographical areas.

Taking the commercialisation process global is another possible direction. It is possible to localise the software to fit the demands of students worldwide, as the issues experienced by university students are not specific to any one nation or area. This would entail local data protection rules being followed, the app being translated into many languages, and the content being modified to account for cultural variances.

In conclusion, the commercialization of the social chat application presents a significant opportunity to make a positive impact on university students' lives while also generating revenue through a sustainable business model. By leveraging digital marketing, establishing strategic partnerships, and adhering to legal and ethical standards, the app can achieve widespread adoption and contribute to improving the social well-being and academic success of students worldwide

## 13. Conclusion

The substantial effects that relationship problems—especially those involving friends and family—may have on university students' general well-being and academic achievement have been brought to light by this research. This social chat application was created to improve student's social interactions and foster their personal development, after it became clear that there was a need for practical solution to these problems. Through the use of cutting-edge technologies like React, Zustand, Dailogflow and Firebase the application provides students with an easy-to-use platform on which they can participate in meaningful conversations, get feedback on their social interactions and take proactive measures to enhance their friendliness, open-mindedness and communication skills.

A key component of the program is the peer-based rating system, which gives users the ability to evaluate their social skills and shortcomings, promoting self-awareness and personal growth. The technology allows users to monitor their development over time and strive towards becoming more proficient communicators by offering real-time feedback and securely preserving these ratings. By guaranteeing that students have control over their interactions and can defend themselves against unsolicited communication, the blocking tool further improves user safety.

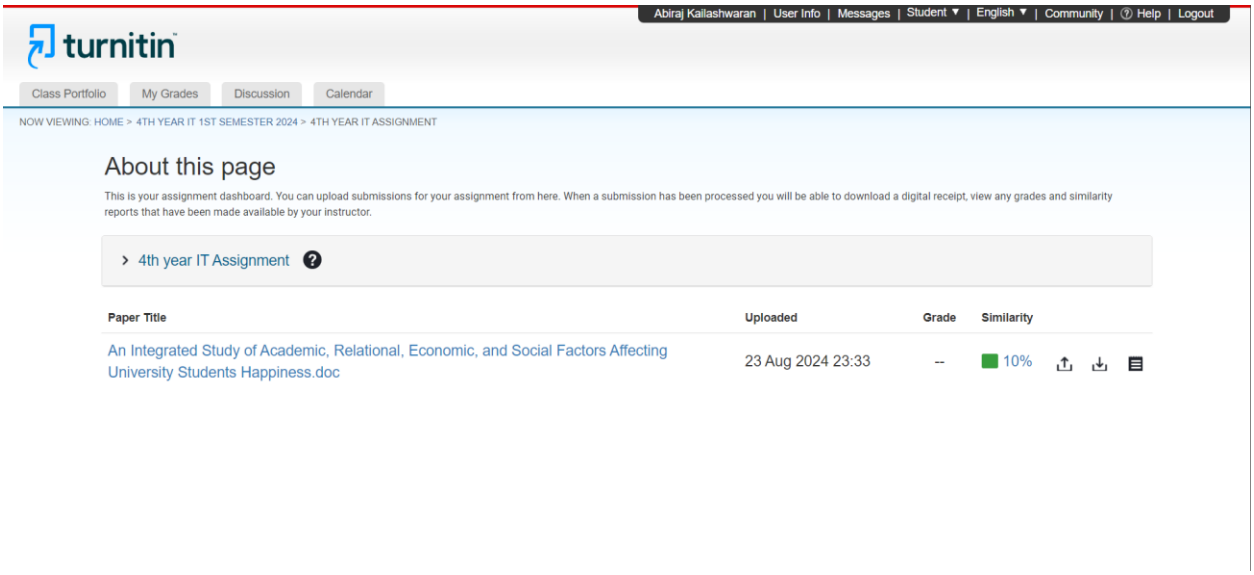
The system's architecture was carefully designed to ensure scalability, security, and ease of use, making it a robust solution for the complex challenges faced by university students in managing their social relationships. While the application has shown promise in improving social interactions and, by extension, academic performance, there is still room for further research and enhancement. Future improvements could include integrating advanced features, such as a chatbot that analyzes user ratings and provides personalized advice on how to improve social skills.

To sum up, this social chat software is a significant step towards fixing the interpersonal problems that commonly compromise the academic success and happiness of college students. Its platform encourages healthy social engagement and personal growth, which can positively affect students' academic achievement and general well-being. Continued feature expansion and enhancement will further increase the use of the program, making it an invaluable resource for college students juggling the pressures of the classroom.

**References:**


- Asher, S. R., & Weeks, M. S. (2014). Loneliness and belongingness in the college years.
- Fernando, S., & Karunaratne, S. (2015). Challenges faced by university students in Sri Lanka: A case study.
- Ghafournia, N. (2014). The relationship between family functioning and academic achievement.
- Jayawardena, P., & Kumari, A. (2019). The impact of academic stress on students' performance in higher education.
- Perera, H. (2018). Financial stress and its impact on academic performance: Evidence from Sri Lankan universities.
- Perera, M., & Kathriarachchi, S. (2020). Parental expectations and academic pressure in Sri Lankan families.
- Roberts, R., Golding, J., Towell, T., & Weinreb, I. (2000). The effects of financial circumstances on the mental and physical health of students.
- Tinto, V. (1993). Leaving college: Rethinking the causes and cures of student attrition.
- Wentzel, K. R. (2017). Peer relationships, motivation, and academic performance at school.
- Wickramasinghe, V., & Samarasekara, N. (2011). Factors affecting university students' dropout decisions: The case of Sri Lanka.

# APPENDICES


Abiraj Kailashwaran | User Info | Messages | Student | English | Community | Help | Logout
turnitin
Class Portfolio
My Grades
Discussion
Calendar
NOW VIEWING: HOME > 4TH YEAR IT 1ST SEMESTER 2024 > 4TH YEAR IT ASSIGNMENT
About this page
This is your assignment dashboard. You can upload submissions for your assignment from here. When a submission has been processed you will be able to download a digital receipt, view any grades and similarity reports that have been made available by your instructor.
4th year IT Assignment
Paper Title
Uploaded
Grade
Similarity
An Integrated Study of Academic, Relational, Economic, and Social Factors Affecting University Students Happiness.doc
23 Aug 2024 23:33
--
10%