

**SOCIAL MEDIA PLATFORM FOR COLLEGE STUDENTS USING C PROGRAMMING LANGUAGE**

**A PROJECT REPORT**

**Submitted by**

***T. Hemanth Kumar (192211935)***

***S. Nithish (192210322)***

*Under the guidance of*

**Dr R DHANALAKSHMI**

*in partial fulfilment for the completion of course*

**CSA0279 – C PROGRAMMING FOR BRGINNERS**

**TITLE**:

**Social Media Platform For College Students**

**PROBLEM STATEMENT:**

In today's digital age, college students face challenges in connecting, collaborating, and sharing information within their academic and social communities. Traditional social media platforms are often too broad and lack the specificity to cater to the unique needs of students, such as course discussions, event planning, resource sharing, and building meaningful connections within their institution. This results in missed opportunities for networking, inefficient communication, and limited access to relevant academic and social resources.

**TASKS :**

* Automatically categorize posts and interactions based on predefined tags (e.g., academics, events, clubs, career opportunities).
* Route specific content or queries to relevant groups or administrators (e.g., academic advisors, club leaders, event coordinators).
* Provide real-time notifications and updates for events, group discussions, or academic resources tailored to individual preferences.
* Enable feedback and polls for events, group activities, or shared content to foster engagement and improve the platform experience.

**OUTCOME :**

The outcome of this project is to develop an automated Social Media Platform for College Students that facilitates seamless communication, collaboration, and resource sharing within the college community. The platform will enable students to share updates, participate in discussions, and access resources while automatically categorizing posts based on predefined criteria (e.g., academics, events, clubs). It will direct queries and interactions to the appropriate groups or personnel, provide real-time notifications, and track engagement. Additionally, it will allow for feedback collection on shared resources, events, and activities to foster continuous improvement.

**AIM**

The aim of this project is to develop an automated Social Media Platform for College Students that facilitates communication, collaboration, and resource sharing among students, faculty, and campus organizations. The platform will categorize posts, queries, and discussions based on predefined topics (e.g., academics, events, clubs), assign them to relevant groups or individuals for responses, and track engagement or resolutions. It will also enable feedback collection on events, shared resources, and activities to ensure continuous improvement.

**ABSTRACT**

The Social Media Platform for College Students is a C-based application designed to foster communication, collaboration, and community-building among students. This system addresses the increasing need for a dedicated platform tailored to the academic and social interactions within educational institutions. Built using structured programming principles, the application offers a streamlined and secure environment for students to connect, share, and engage.

The platform allows users to create profiles, post updates, and interact with peers through comments and likes, promoting a sense of community. Features such as group creation for academic discussions, event announcements, and resource sharing are included to support both social and educational needs. Each user is provided with a unique ID to maintain a personalized experience, and activities are tracked to ensure seamless interaction and content management.

The application leverages C programming features such as file handling for data storage and retrieval, ensuring a lightweight and efficient solution. By automating key aspects of social media interaction, the platform reduces manual coordination efforts and enhances connectivity among students.

**INTRODUCTION**

In the digital era, fostering effective communication and collaboration among college students is crucial for enhancing academic engagement, social interaction, and community building. Traditional communication channels often fail to meet the dynamic and specific needs of students, resulting in missed opportunities for collaboration and limited connectivity. A dedicated social media platform tailored to college environments can address these challenges, creating a unified space for students to connect, share, and grow.

The Social Media Platform for College Students is a software application developed in C programming to cater to these needs. This system provides a structured platform for students to interact, share updates, and participate in academic and extracurricular discussions. Leveraging C’s features such as file handling and structured programming, the application ensures an efficient and lightweight solution for campus-based social networking.

Key functionalities of the platform include:

• Enabling students to create and manage personal profiles.

• Allowing users to post updates, share resources, and comment.

• Creating groups for academic discussions and event planning.

• Tracking user activities and interactions for a seamless experience.

• Providing a secure and focused environment tailored to college students.

By automating key aspects of social media interaction, the platform reduces manual coordination, enhances student engagement, and fosters a sense of community. While currently a standalone application, it forms the foundation for more advanced systems integrating multimedia capabilities, databases, or web-based interfaces.

This project highlights the practical implementation of C programming in addressing real-world challenges, showcasing the potential of software to improve student connectivity and collaboration within educational institutions.

**CODE IMPLEMENTATION**

**Modules:**

**1. User Registration and Login:** Enables students to create an account and log in to the platform.

* a. User Details: Collects user information, such as name, email, password, and college details.
* b. Login Validation: Ensures only registered users can access the platform by verifying credentials.
* c. Data Saving: User data is stored securely in a structured text file (e.g., users.txt).

**2. Profile Management:** Allows students to manage and update their profiles.

* a. Edit Profile: Enables users to update details like bio, profile picture (text-based representation), and interests.
* b. View Profile: Displays the user’s profile information to others on the platform.

**3. Post Creation and Interaction:** Facilitates content sharing and peer interaction.

* a. Create Post: Users can write posts, share ideas, or upload resources (e.g., links).
* b. Comment and Like: Users can comment on or like posts to encourage interaction.
* c. Data Saving: Posts and interactions are stored in a structured file (e.g., posts.txt).

**4. Group and Event Management: Supports** collaboration and organization of student activities.

* a. Create Group: Users can create groups for academic or extracurricular discussions.
* b. Join/Leave Group: Users can join or leave groups as needed.
* c. Event Posting: Organizers can post events with details like date, time, and location.

**5. Activity Feed:** Displays updates from peers and groups in chronological order.

* a. Feed Generation: Aggregates posts, comments, and event updates.
* b. Filter Options: Users can filter the feed by group or type of content.

**6. Messaging System:** Allows private communication between students.

* a. Send Message: Users can send direct messages to other students.
* b. Inbox Management: Displays received messages in a structured format.

**7. Data Storage:**

* Use text files (e.g., users.txt, posts.txt, groups.txt, messages.txt) to store data.
* Use structured formats like CSV for easy parsing and retrieval.

**Flow:**

* 1. Main menu for user interaction.
* 2. Submenus for different functionalities (e.g., Profile, Posts, Groups).
* **Code Outline:**
* 1. Struct Definition:
* 2. Functions:

a. User Registration and Login:

b. Profile Management:

c. Post Management:

d. Group and Event Management:

e. Messaging:

* 3. Main Menu:

Implement a main menu to navigate between modules.

**PROGRAM**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#define MAX\_USERS 100

#define MAX\_POSTS 100

#define MAX\_LENGTH 256

typedef struct {

int id;

char name[MAX\_LENGTH];

char email[MAX\_LENGTH];

char password[MAX\_LENGTH];

} User;

typedef struct {

int id;

int userId;

char content[MAX\_LENGTH];

char timestamp[MAX\_LENGTH];

} Post;

void registerUser(User users[], int \*userCount) {

printf("\n--- Register User ---\n");

if (\*userCount >= MAX\_USERS) {

printf("Error: User limit reached.\n");

return;

}

users[\*userCount].id = \*userCount + 1;

printf("Enter Name: ");

scanf(" %[^\n]", users[\*userCount].name);

printf("Enter Email: ");

scanf(" %[^\n]", users[\*userCount].email);

printf("Enter Password: ");

scanf(" %[^\n]", users[\*userCount].password);

(\*userCount)++;

printf("User Registered Successfully!\n");

}

int loginUser(User users[], int userCount) {

char email[MAX\_LENGTH], password[MAX\_LENGTH];

printf("\n--- User Login ---\n");

printf("Enter Email: ");

scanf(" %[^\n]", email);

printf("Enter Password: ");

scanf(" %[^\n]", password);

for (int i = 0; i < userCount; i++) {

if (strcmp(users[i].email, email) == 0 && strcmp(users[i].password, password) == 0) {

printf("Login Successful! Welcome, %s\n", users[i].name);

return users[i].id;

}

}

printf("Invalid email or password. Please try again.\n");

return -1;

}

void createPost(Post posts[], int \*postCount, int userId) {

printf("\n--- Create Post ---\n");

if (\*postCount >= MAX\_POSTS) {

printf("Error: Post limit reached.\n");

return;

}

posts[\*postCount].id = \*postCount + 1;

posts[\*postCount].userId = userId;

printf("Enter Post Content: ");

scanf(" %[^\n]", posts[\*postCount].content);

printf("Enter Timestamp (e.g., 2024-12-02 14:30): ");

scanf(" %[^\n]", posts[\*postCount].timestamp);

(\*postCount)++;

printf("Post Created Successfully!\n");

}

void viewPosts(Post posts[], int postCount, User users[], int userCount) {

printf("\n--- View All Posts ---\n");

if (postCount == 0) {

printf("No posts available.\n");

return;

}

for (int i = 0; i < postCount; i++) {

printf("Post ID: %d\n", posts[i].id);

for (int j = 0; j < userCount; j++) {

if (users[j].id == posts[i].userId) {

printf("Posted By: %s\n", users[j].name);

break;

}

}

printf("Content: %s\n", posts[i].content);

printf("Timestamp: %s\n", posts[i].timestamp);

printf("-------------------------\n");

}

}

int main() {

User users[MAX\_USERS];

Post posts[MAX\_POSTS];

int userCount = 0, postCount = 0;

int loggedInUser = -1;

int choice;

while (1) {

printf("\n--- Social Media Platform for College Students ---\n");

if (loggedInUser == -1) {

printf("1. Register\n");

printf("2. Login\n");

} else {

printf("1. Create Post\n");

printf("2. View Posts\n");

}

printf("3. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

if (loggedInUser == -1) {

switch (choice) {

case 1:

registerUser(users, &userCount);

break;

case 2:

loggedInUser = loginUser(users, userCount);

break;

case 3:

printf("Exiting the system.\n");

exit(0);

default:

printf("Invalid choice. Please try again.\n");

}

} else {

switch (choice) {

case 1:

createPost(posts, &postCount, loggedInUser);

break;

case 2:

viewPosts(posts, postCount, users, userCount);

break;

case 3:

printf("Exiting the system.\n");

exit(0);

default:

printf("Invalid choice. Please try again.\n");

}

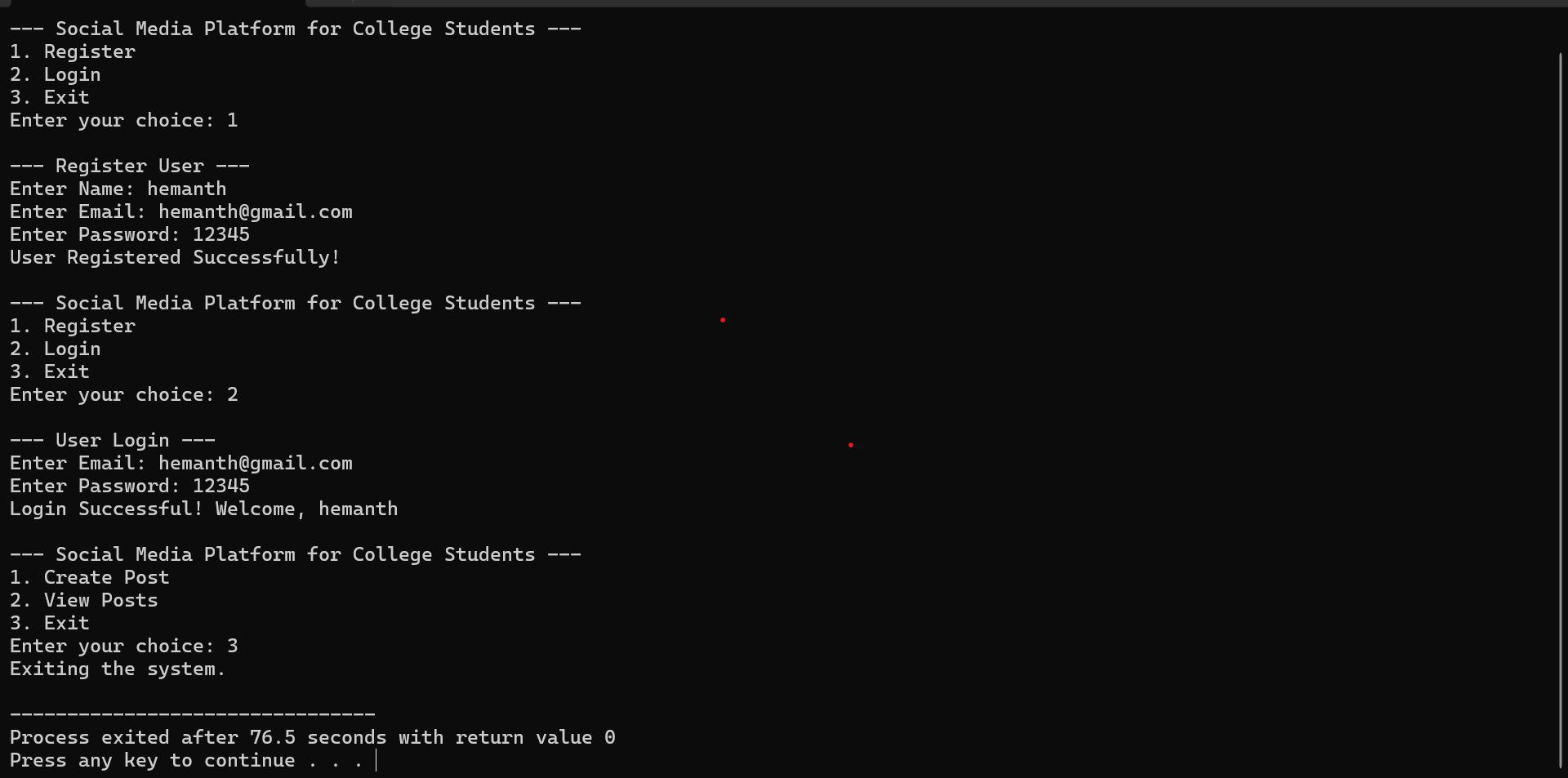
}

}

return 0;

}

**RESULT**

****

**ENGINEERING STANDARDS**

C Engineering Standards for a Social Media Platform for College Students

Developing a social media platform for college students requires adherence to a range of engineering standards to ensure functionality, security, usability, and scalability. Below are the key standards categorized to address the specific needs of such a platform:

**1. Software Development Standards**

**I. ISO/IEC 12207:** Software Development Life Cycle (SDLC) Processes

Provides a systematic framework for planning, developing, testing, deploying, and maintaining the platform.

Ensures thorough documentation and organization at each phase of development.

**II. IEEE 829:** Software Testing Documentation

Guides the creation of detailed testing documentation, including test cases, plans, and reports.

Helps validate the platform’s functionality, security, and usability before deployment.

**III. ISO/IEC 25010:** Software Quality Model

Functionality: Ensures the platform meets features like profile creation, messaging, and group management.

Reliability: Guarantees consistent operation under heavy user traffic.

Usability: Designs user-friendly interfaces suitable for college students.

Performance Efficiency: Optimizes response times, particularly during peak usage.

Security: Protects user data and interactions from breaches.

Maintainability: Allows for easy updates and feature additions.

Portability: Ensures the platform is accessible on various devices and operating systems.

**IV. IEEE 730:** Software Quality Assurance

Emphasizes the importance of reviews, audits, and evaluations to maintain quality throughout development.

**2. C Language Standards** (If Using C for Development)

If parts of the platform are built using C (e.g., for backend processes or algorithms):

**ISO/IEC 9899:2011 (C11):** Ensures modern features like multithreading and bounds-checking functions for secure operations.

**ISO/IEC 9899:2023 (C23):** Incorporates the latest features for better performance, usability, and compatibility.

**3. Data Security and Privacy Standards**

**I. ISO/IEC 27001:** Information Security Management

Defines requirements for a robust information security management system to protect user data.

**II. GDPR and CCPA Compliance**

Ensures adherence to data privacy laws applicable to users in different regions.

Provides transparency regarding data collection, storage, and usage.

**III. OWASP Standards**

Implements best practices to safeguard the platform against common security threats like SQL injection, XSS, and CSRF.

**4. Usability and Accessibility Standards**

I. ISO 9241: Ergonomics of Human-System Interaction

Focuses on creating an intuitive, user-friendly interface for seamless navigation.

II. WCAG (Web Content Accessibility Guidelines)

Ensures the platform is accessible to users with disabilities, including those relying on assistive technologies.

III. ISO/IEC 25062: Common Industry Format for Usability Test Reports

Guides usability testing and reporting to enhance the user experience.

**5. System Performance Standards**

I. ISO/IEC 14764: Software Maintenance

Ensures that the platform remains reliable and performant through updates and maintenance.

II. ISO/IEC 9126: Software Engineering – Product Quality

Sets benchmarks for performance efficiency, including response times and resource optimization.

By adhering to these engineering standards, the social media platform for college students will provide a secure, user-friendly, and scalable solution tailored to meet both current needs and future demands.

**FUTURE SCOPE**

A social media platform for college students can evolve into a robust and dynamic tool to foster connections, collaboration, and opportunities within the academic ecosystem. Here are potential future advancements for such a platform:

**1. Database Integration and Optimization**

Implement advanced database systems (e.g., MongoDB, PostgreSQL) to handle growing user data efficiently.

Ensure scalability to support increased user activity, media uploads, and interactions.

**2. Web and Mobile Accessibility**

Develop user-friendly web and mobile applications for seamless access.

Use frameworks like React, Flutter, or Swift to ensure compatibility across devices and platforms.

**3. Content Personalization**

Employ machine learning algorithms to recommend content, events, groups, and connections based on users’ interests and behavior.

Introduce personalized dashboards for academic resources, clubs, and events.

**4. Enhanced Communication Tools**

Add real-time chat, video conferencing, and group collaboration features.

Integrate tools for academic discussions, such as whiteboards and document sharing.

**5. Event Management and Tracking**

Enable students to create, share, and RSVP to college events with real-time updates.

Include features for virtual event hosting and live streaming.

**6. Gamification Features**

Introduce gamified elements like badges, leaderboards, and reward points for participation in discussions, quizzes, or events.

Encourage engagement by rewarding active users with perks like access to exclusive content.

**7. Advanced Networking Opportunities**

Facilitate professional networking by integrating features like mentor matching, alumni connections, and internship/job boards.

Partner with LinkedIn or similar platforms for broader career opportunities.

**8. AI-Powered Moderation**

Leverage AI to detect and filter inappropriate content, ensuring a safe and respectful environment.

Use sentiment analysis to monitor the platform's overall tone and address concerns promptly.

**9. Multi-Language and Accessibility Support**

Add multi-language support to cater to diverse user groups in multicultural campuses.

Ensure accessibility compliance (e.g., WCAG) to accommodate users with disabilities.

**10. Analytics and Insights**

Provide insights into user engagement, popular topics, and feedback for administrators to enhance the platform’s value.

Use analytics to recommend strategies for better engagement and resource allocation.

**CONCLUSION**

The Social Media Platform for College Students, developed using C programming, serves as a foundational application for fostering communication, collaboration, and community-building within educational institutions. By enabling user registration, profile management, post creation, and interaction, the platform addresses key challenges in student connectivity and engagement, providing a dedicated space for academic and social networking.

This project demonstrates the practical application of C programming in solving real-world challenges, utilizing its file handling and structured programming capabilities. While the current implementation focuses on basic functionalities, it highlights the potential of technology in enhancing student interactions, reducing communication barriers, and promoting a sense of community among college students.

The system’s potential for future enhancements, such as incorporating group discussions, event management, multimedia features, and integration with databases or web interfaces, ensures its scalability and relevance in modern educational environments. With further development, the platform can evolve into a comprehensive tool that significantly contributes to student collaboration, learning, and engagement.