
# DATA STRUCTURES AND ITS APPLICATIONS (UE22CS252A)

## Mini Project

## Project Title & Team Members

### Title

# SOCIAL MEDIA NETWORKING

### Team Members (SRN Name)

1. NISHANT VITHAL GHORPADE
2. N C SAI SIDDHARTHA
3. NIRANJAN MANJUNATH NAIK
4. NIRMITH MN

## Synopsis


**Idea:**

**The primary idea is to create a platform where users can register, connect with friends, share updates, view posts from friends, and manage their posts. The system utilizes data structures such as structures and linked lists to represent users, posts, and the relationships between users. It employs a menu-driven interface for user interaction.**

- **Allows user registration, login, friend addition, post updates, and viewing posts/friends.**
- **Data structures: User, Node, Graph (adjacency list), and Post.**
- **User registration and login with basic authentication.**
- **Adding friends updates adjacency lists with mutual connections.**
- **Posting updates with content limited to 500 characters.**
- **Viewing a personalized news feed with user and friend posts.**
- **Continuous loop presents a menu for user interaction.**

# Abstract Data Types


**User:**

- **Attributes: id, username, password**
- **Represents a user in the social network.**

**Graph:**

- **Attributes: user (of type User), friends (a linked list of Node structures)**
- **Represents the social network as an array of Graph structures. Each Graph structure corresponds to a user and contains information about the user and their friends.**

**Node:**

- **Attributes: userId, next (a pointer to the next Node)**
- **Represents a node in the linked list used for the adjacency list of friends. Each Graph structure has a linked list of friends represented by Node structures.**

**Post:**

- **Attributes: id, authorId, content**
- **Represents a post in the social network, including the post's ID, the ID of the user who authored it (authorId), and the content of the post.**

# Functionalities

**User Registration (registerUser):**

- **Users can register by providing a username and password.**
- **The program checks if the maximum user limit has been reached.**

**User Login (login):**

- **Users can log in by entering their username and password.**
- **The program verifies the credentials and logs in the user if they are valid.**

**Add Friend (addFriend):**

- **Users who are logged in can add other users as friends.**
- **The program checks if the users are not already friends before creating a connection.**

**View Friends (viewFriends):**

- **Logged-in users can view their list of friends.**
- **The program displays the usernames of the logged-in user's friends.**

**Post Update (postUpdate):**

- **Logged-in users can post updates (limited to 500 characters).**
- **The program adds the post to the global array of posts.**

# Functionalities

**View Post (viewPost):**

- **Logged-in users can view their news feed, consisting of their own posts and posts from friends.**
- **The program displays the content of each post, indicating the author.**

**Menu-Driven Interface (main):**

- **The main function provides a menu for users to choose from various actions.**
- **Users can register, log in, add friends, post updates, view posts, view friends, or exit the program.**
- **The program runs in a loop, allowing users to perform multiple actions in a single session.**

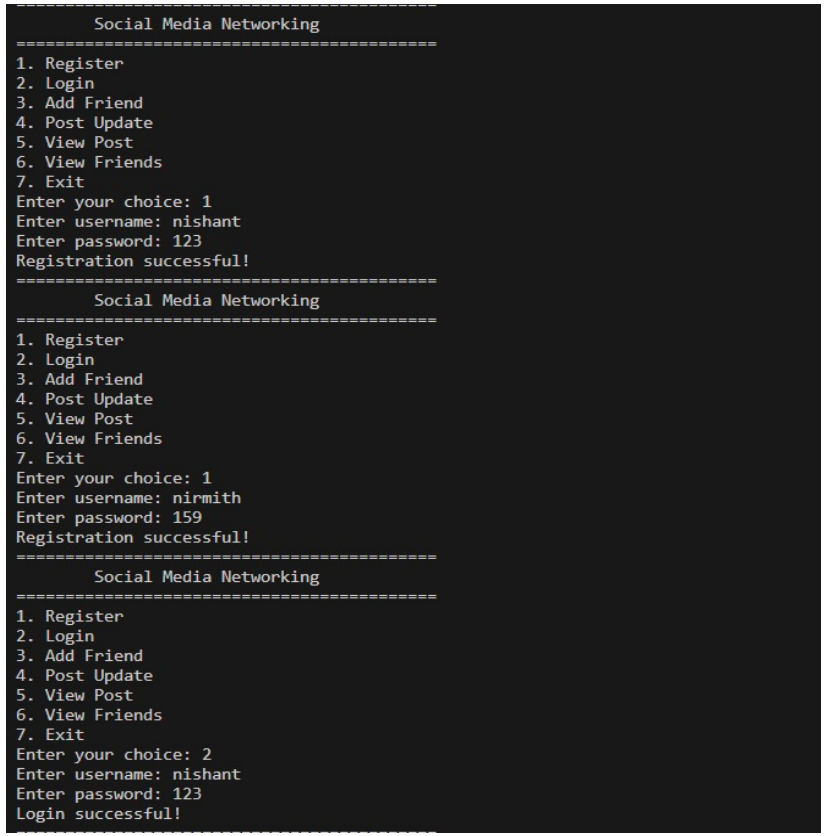
**Data Structures (User, Node, Graph, Post):**

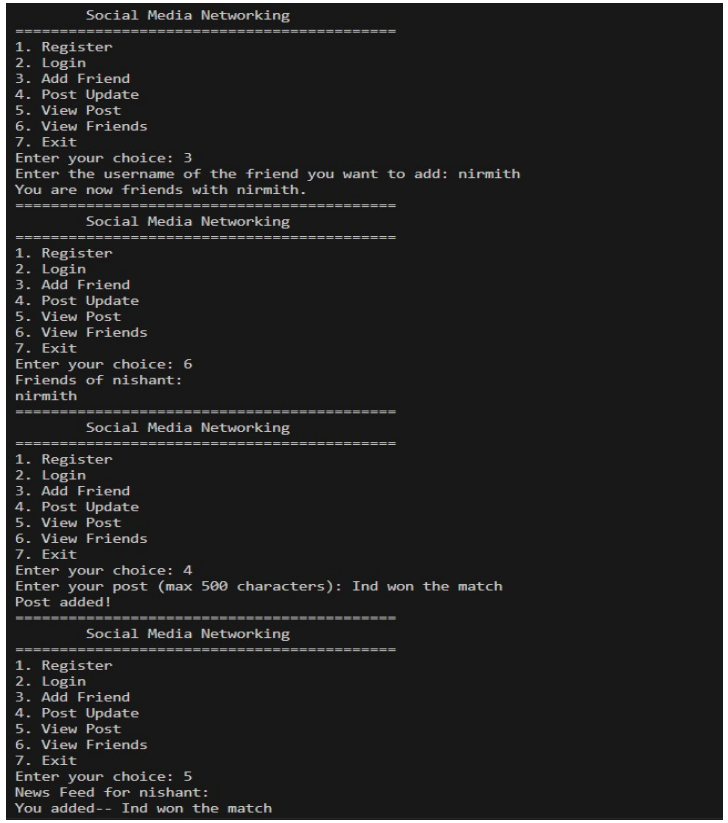
- **The program uses structured data to represent users, friends (in an adjacency list), and posts.**

**Dynamic Memory Allocation:**

- **Memory is allocated dynamically using malloc when adding friends to the adjacency list.**

# Interface - Code  Output



## Contribution of each Team Member


**Nishanth Vitthal Ghorphade - PES2UG22CS368 :**

**Register User( ) and Login( ) funtion**

**Niranjan Manjunath Naik - PES2UG22CS364 :**

**Add post( ) and view post( ) function**

**Nirmith M N - PES2UG22CS365:**

**Add friends( ) and display friends ( ) function**

**N C Sai Siddhartha - PES2UG22CS326 :**

**Declaration of all data structure , user index ( ) and delete post() function**