B.Tech, Sem III
Session : Aug-Dec, 2019

UE18CS202 – Data Structures

REPORT
ON
"SOCIAL NETWORK ANALYSIS"

SECTION : J

#	SRN	Name	Contact No.	Email ID	Sign
1	PES1201802828	Himanshu Jain	7848003517	nhimanshujain@gmail.com	
2	PES1201801890	Rohit Kumar A.S	9449569502	natsuanime24@gmail.com	

ABSTRACT

" Social Network Analysis " is a representation of the way different features are implemented in Social networking sites. It attempts to understand and represent how users and their data is handled, how relationships between users are handled.

The Data structures used are Directed graphs, using Adjacency list implementation. **The Abstract Data Structure has the following prototypes :-**

1 **Insert_front**: This takes 1 argument from the current user ; the request id, and 2 other arguments internally ; the user_id and the request's name. It creates a link between the current user and his "friend". It has no return value.

2 **Display** : It takes the user_id and the data structure's address as input and prints the details about the current user and his friends. It has no return value.

3 **Create** : It is an internally used function, that takes name and id as inputs and returns a node containing the user data

4 **Read** : It is an internally used function that has no arguments but rather reads data from the .csv file . It analyses the read data and creates the appropriate graph.

➤ Advantages

1. Small scale network can be maintained easily .

2. Social relationships can be modeled and represented online.

3. Since this is basically an attempt at mimicking the real-world social networking sites, this gives knowledge into how real-world sites are constructed and operated.

DESCRIPTION OF DATA STRUCTURE, LOGIC AND FUNCTIONALITY

➤ Description of Data Structure :

1) Data structure used is the *directed graph implemented using adjacency list*. The adjacency list is an array of pointers to a structures called *'user'*

2) "User" is a structure that stores data about the user. It has ;
   ```
   {
      int no_of_friends;
      int id;
      char name[50];
      struct user *link;
   }
   ```

3) The array of pointers is called *"all_user"*.

➤ Logic

1) Every user in the social network is represented as a *node* .

2) This node stores all the required information about a user such as the unique id, name, number of friends .

3) The unique id is also used to login so that the user will have an option to display the list of all his friends

➤ Functionality and Implementation

1. Sign-up :

A new user signs up by entering his name. We then verify for existing user in the network by checking the name entered ;by traversing through all the nodes, using "*strcmp*". If a duplicate name is found, a message is displayed asking him to login instead of signing in. Else, a unique id is generated based on the sequence of the array and a node is dynamically created using the "create" function .

2. Login :

A user with an existing account, logs in by entering his id. If an invalid id is entered, login is prohibited to the user.

3. Making Friends :

A logged-in user can send a friend request to an existing user if he knows his username. When this happens, the "insert_front" function is called. This function verifies the log-in attempt; if the requested user exists, then that node is inserted at the front of the list of the current user.

4. Display :

Every account created has a display option , which displays the details about the user. Essentially, this prints the user's name, id, number of friends and the list of friends he/she has. The list of friends is obtained by traversing through the linked list of the user using a temporary "temp" pointer.

CONCLUSION

The project has given us insights into how society is modeled and social relationships are represented in the online world to create an immersive and scalable experience through social networking. Knowledge about the representation of users/entities on the networking sites and on some of the features, was gained.

The project has large potential that can be tapped to make it large-scale and implement unique and immersive features that might not be viable for mammoth real-world social networking sites but suitable for small-scale university-level sites. Our college could definitely benefit from one.

- ➢ Account handling (sign-in, login) can be made more secure and authorised.
- ➢ Linking this with servers can give rise to simultaneous multi-device usage.
- ➢ An online discussion/ forum can be implemented in the future to make it more immersive.