**APPLICATION WALK THRU WITH mySQL QUERIES**

**Option 1 Existing User:**

The first page or landing page would be LOGIN PAGE with an option to create new user

Login Screen

User Name [ ]

Password [ ]

[SUBMIT]

New User Registration

If it is existing user the system will ask to put the username ans password if it is a match then it would enter the main page otherwise it will give 10 chances to enter into the system otherwise it will go to exit page.

The query for the checking the user name and the password would be**…(Query 1)**

**SELECT id, userEmail, userPwd from tbluser where userPwd = inPwd and userEmail = inUserName;**

(the password is not encrypted for now but later we can add this feature if we have time)

If matched then redirect to main page with userId and Name and other details with url

Otherwise warn user for 10 times and then redirect to **Not found page.**

**Option 2 New User:**

If New User the user will click the new user link which would take the user to New User Page where user will enter the details of the new user.

New User

\*User Email [ ]

\*Name [ ]

Address [ ]

\*Password [ ]

\*Contact Number [ ]

[SUBMIT]

\*Are required entries

Query 2

Inserting the details for the New User

**INSERT INTO tbluser (userName, userAddress, userEmail, userPwd)**

**SELECT \* FROM (SELECT In\_userName,** **In\_userAddress, In\_userEmail, In\_userPwd)**

**AS tmp WHERE NOT EXISTS ( SELECT userEmail FROM tbluser WHERE userEmail = In\_userEmail) LIMIT 1;**

Along with insert, another entry needs to be placed in the friends table to make one self as a friend for the first time.

The following query will process after successful insertion of the record.

**INSERT INTO tblfriendlist SET mainUserId = In\_userId, friendUserId = In\_userId;**

(Note: The In\_ prefix is from the user input screen)

**UPDATE tbluser SET username =In\_userName,**

**userAddress=In\_userAddress,userEmail=In\_userEmail,**

**userPwd=In\_userPwd WHERE userId = In\_userId;**

Changing the values of the User

The Column userEmail needs to be unique to prevent any duplicate entries in the table.

**Listings and Insertion Data**

Following listing needs to be populated in main screen

1. **Main query for showing comments from the friend list**

**SELECT distinct a.userId, b.mainUserId, b.friendUserId, c.mainUserId,c.post**

**FROM facefin\_db.tbluser AS A**

**LEFT JOIN tblfriendlist AS B**

**ON A.userId = B.mainUserId**

**INNER JOIN tblpost as C**

**ON B.friendUserId = C.mainUserId and a.userId=In\_userId**

**order by c.tstamp DESC LIMIT 10;**

1. **Query for Listing top 5 friend’s suggestion**

**Select b.userName from tblfriendlist AS A INNER JOIN tbluser AS B ON userId=A.id AND**

**NOT mainUserId=In\_userId AND NOT friendUserId = In\_userId**

**OR NOT acceptFrom= In\_userId AND NOT acceptTo= In\_userId**

**AND NOT requestFrom= In\_userId**

**AND NOT requestTo= In\_userId LIMIT 5;**

1. **Query for listing top 5 invitations**

**Select userName from tblfriendlist**

**INNER JOIN tbluser where requestTo = In\_userId**

**and acceptTo=0 and mainUserId=userId;**

1. **Inserting New Posts**

**insert into tblpost set mainUserId = In\_userId,**

**post =In\_Post;**

1. **Sending Invitation**

In the Listing (2) **Top five friends** click the button to send the invitation to that friend

**INSERT INTO tblfriendlist SET**

**mainUserId = In\_userId,**

**friendUserId=****In\_sentTo,**

**requestTo= In\_sentTo,**

**requestFrom = In\_userId;**

1. **Accepting Invitation**

For accepting the invitation there would be two entries in the table and table for showing the invitation will be refreshed (3)

1. Entry would update of acceptFrom and acceptTo columns in the tblfriendlist
2. Entry would be Insert into tblfriendlist filling up all the columns reversing the mainuserId position and requestFrom columns this need to be done in order to create new entry for the accepting user.

Ex: 3 sent Invitation to 5

**Update tblfriendlist SET acceptFrom = 3, acceptTo = 5 where mainUserId = 3 and acceptTo = 0;**

**insert into tblfriendlist set mainUserId = 5, friendUserId = 3, requestTo = 3, requestFrom =5,acceptFrom = 3, acceptTo = 3;**