FriendzClub

Social Network Friend Recommendation System

Project File for DS
JavaFx Project

Prepared By:
FA23-BCS-B-116(Muhammad Hasnat Rasool)
Submitted To:
Teacher : Mamoona Tasadduq

Table of Contents

Table of Contents

CHAPTER 1

1.1Overview:

- The **Social Network Friend Recommendation System** is an innovative and practical project designed to mimic the social networking features of modern platforms. By utilizing graph theory concepts, this system simulates the dynamics of real-world social networks, allowing users to discover mutual connections and receive friend recommendations. The system serves as an excellent platform for exploring data structures, algorithms, and real-world application design.

  In this system:

- **Users** are represented as **nodes** (or vertices) in a graph.
- **Friendships** are depicted as **edges** connecting these nodes.

  By constructing and traversing the graph, the system provides insights such as identifying mutual friends and recommending new connections. This feature-rich application is ideal for demonstrating practical implementation skills in computer science.

1.2Purpose:

The key objectives of this system are:

- To simulate a real-world social network using graph-based representations.
- To enhance user engagement by suggesting relevant connections (friends-of-friends).
- To offer tools for efficient network analysis, such as identifying mutual friends.
- To provide an intuitive and interactive user experience with friend management features.
- To create a scalable framework for implementing social network functionalities.

1.3 Scope:

The **Social Network Friend Recommendation System** focuses on essential features of a social networking platform. The system includes functionalities like adding and removing friends, recommending new connections, and discovering mutual friends. Additionally, it offers a posting system that simulates a real-time activity feed for users. These features make it a comprehensive project for understanding the application of graph theory in a practical setting.

1.4 **Required Features :**

- ✔ Add Friends
- ✔ Delete Friends
- ✔ Display Mutual Friends
- ✔ Suggest Potential Friends Based on Common Connections
- ✔ User Posting System

1.5 **Future Ideas:**

- Badges , achievements for creating more friend (e.g 1st friend , 50th or 100th friends badge).
- Profile picture feature or avators for personalized profile system.
- Real Time chat messages , groups formation .
- Topic based posts filtering , poll questions , comments on posts.

CHAPTER 2

REQUIREMENT SPECIFICATIONS

2.1Hardware Requirements :

Processor Brand	: Intel
Processor Type	: Core i5,i7
Processor Speed	: 4 GHz
Processor Count	: 1
RAM Size	: 4,8 GB
Memory Technology	: DDR3
Computer Memory Type	: DDR3 SDRAM
Hard Drive Size	: 50 GB

2.2Software Requirements :

Operating system	: Windows 10 – Arch Linux
Front end	: JAVA-FX
Connectivity	: JDBC Driver
Database connectivity	: MYSQL , MariaDB

CHAPTER 3

TOOL DESCRIPTION

3.1 Overview of Front End

An important issue for the development of a project is the selection of suitable front- end and back-end. When we decided to develop the project we went through an extensive study to determine the most suitable platform that suits the needs of the requirements as well as helps in development of the project.

The aspects of our study included the
following factors.

Front-end selection:

1. It must have a graphical user interface that assists users that are not from IT background.
2. Scalability and extensibility.
3. Flexibility.
4. Robustness.
5. According to the requirement and the culture.
6. Platform independent.
7. Easy to debug and maintain.
8. Event driven programming facility.
9. Front end must support some popular back end like MySQL.

According to the above stated features we selected JAVA-FX and CSS as the front-end for developing.

3.1.1 About Java:

Java is a general-purpose, class-based, object-oriented programming language designed for having lesser implementation dependencies. It is a computing platform for application development. Java is fast, secure, and reliable, therefore. It is widely used for developing Java applications in laptops, data centers, game consoles, scientific supercomputers, cell phones, etc.

Here are some important Java applications:

- It is used for developing Android Apps
- Helps you to create Enterprise Software
- Wide range of Mobile java Applications
- Scientific Computing Applications
- Use for Big Data Analytics
- Java Programming of Hardware devices
- Used for Server-Side Technologies like Apache, JBoss, GlassFish, etc.

3.2 Overview of Back End

Back End Selection:

1. Multiple user support.
2. Efficient data handling.
3. Efficient data retrieval and maintenance.
4. Stored procedures.
5. Popularity.
6. Operating System compatible.
7. Easy to install.
8. Various drivers must be available.
9. Easy to implant with the Front-end.

According to above stated features we selected MySQL as the backend.

The technical feasibility is frequently the most difficult area encountered at this stage. It is essential that the process of analysis and definition be conducted in parallel with an assessment to technical feasibility. It centers on the existing computer system (hardware, software etc.) and to what extent it can support the proposed system.

3.2.1 About SQL:

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons.[1]

MySQL is released under an open-source license. So you have nothing to pay to use it. MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages. MySQL uses a standard form of the well-known SQL data language. MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.

MySQL works very quickly and works well even with large data sets. MySQL is very friendly to PHP, the most appreciated language for web development. MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).

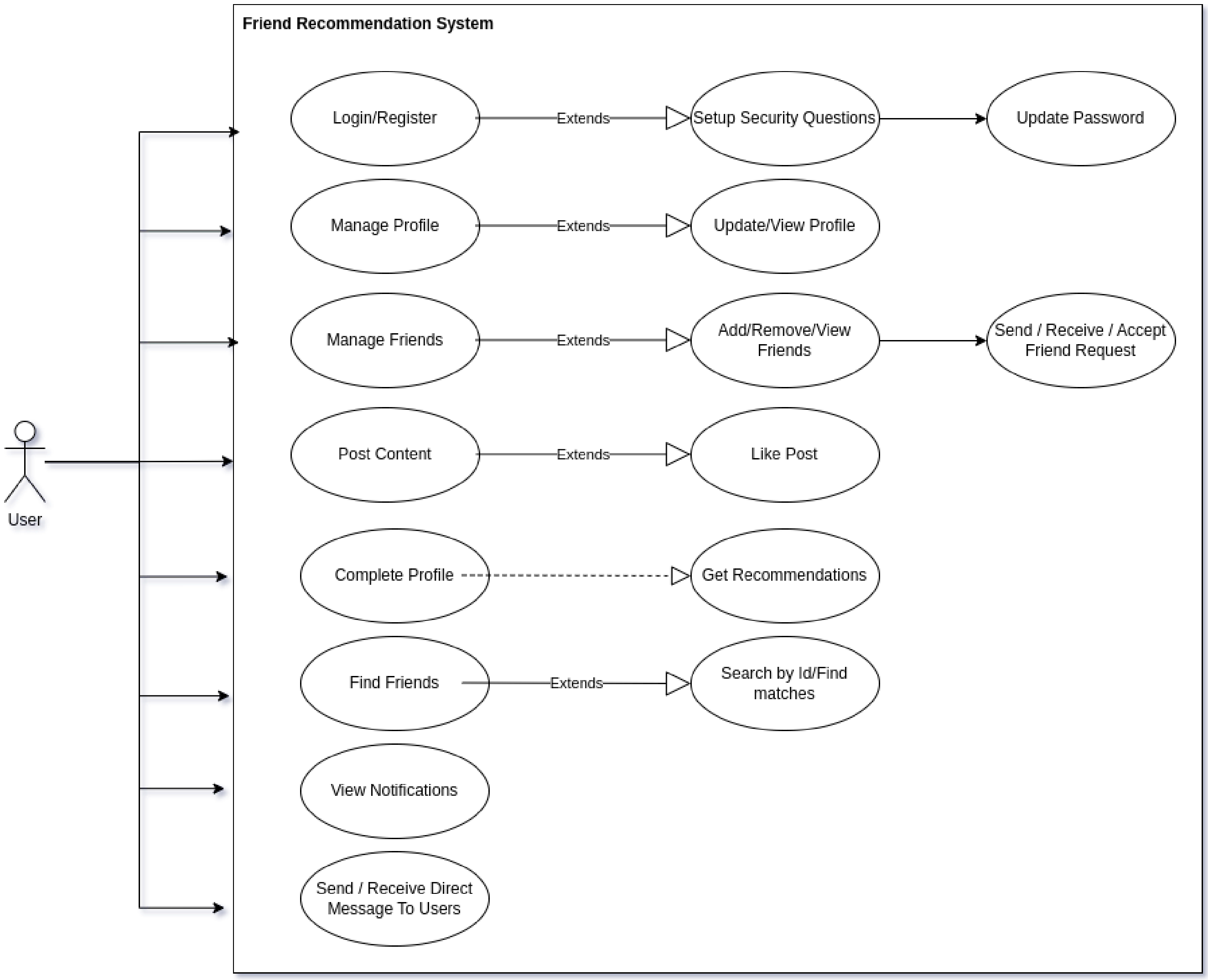
Also, they are using different dialects, such as −

- Oracle using PL/SQL,.
- SQL is widely popular because it offers the following advantages −
- Allows users to access data in the database management systems.
- Allows users to describe the data.relational
- Allows users to define the data in a database and manipulate that data.
- Allows to embed within other languages using SQL modules, libraries & pre-compilers.
- Allows users to create and drop databases and tables.
- Allows users to create view, stored procedure, functions in a database.
- Allows users to set permissions on tables, procedures and views.

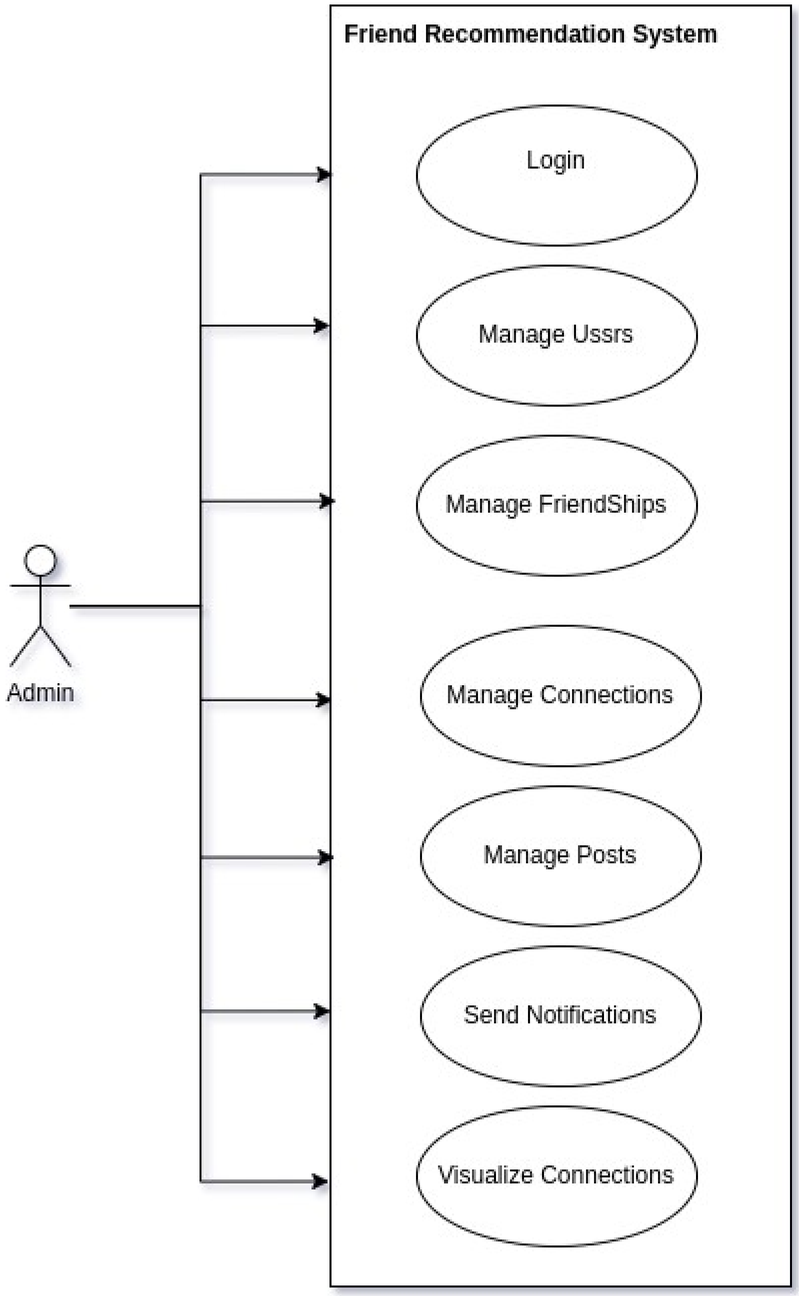
CHAPTER 4

REQUIREMENT ANALYSIS

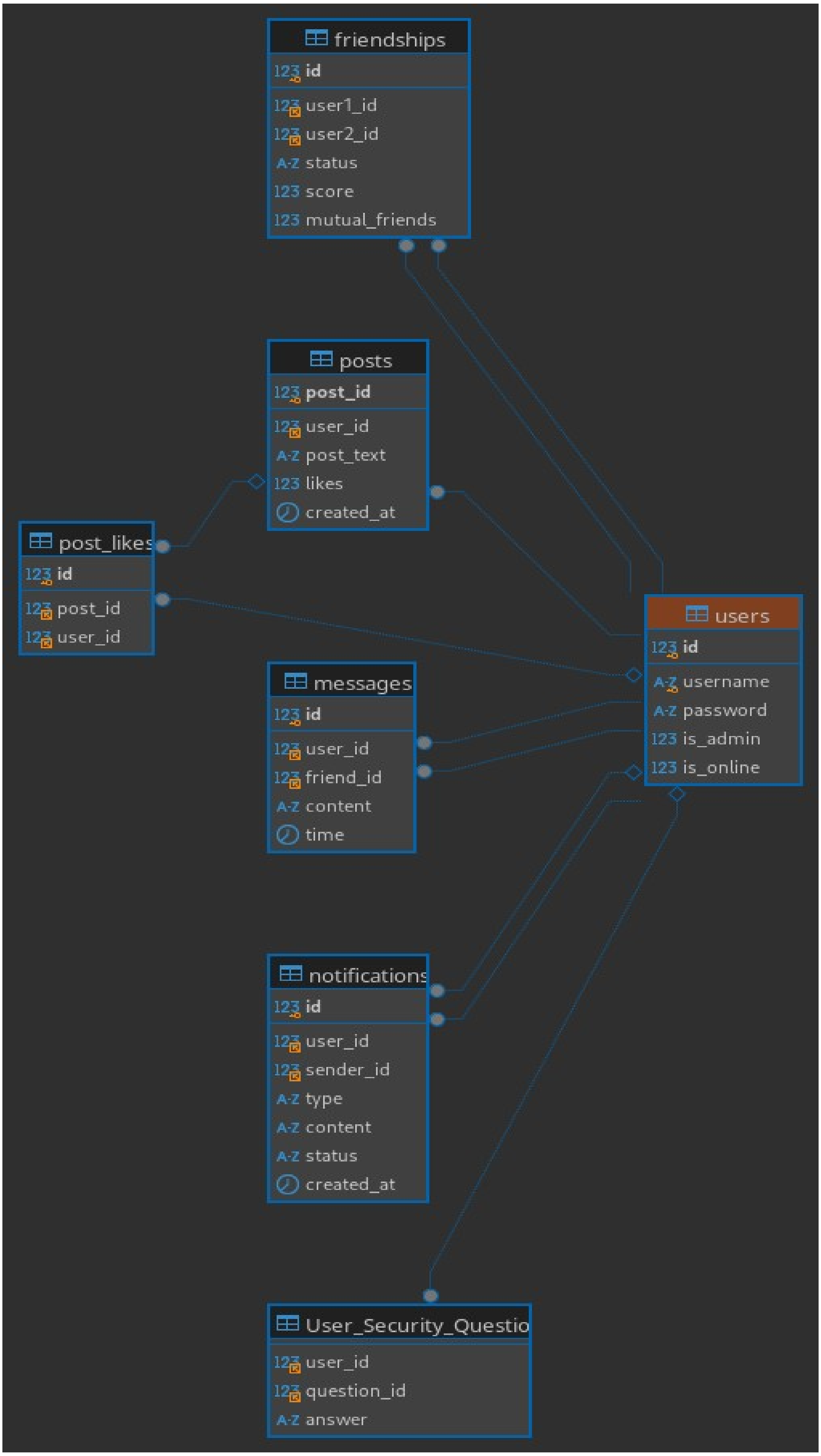
4.1 UseCase Diagram:



4.2 UseCase Diagram:



4.3 Tables Relation:



CHAPTER 5

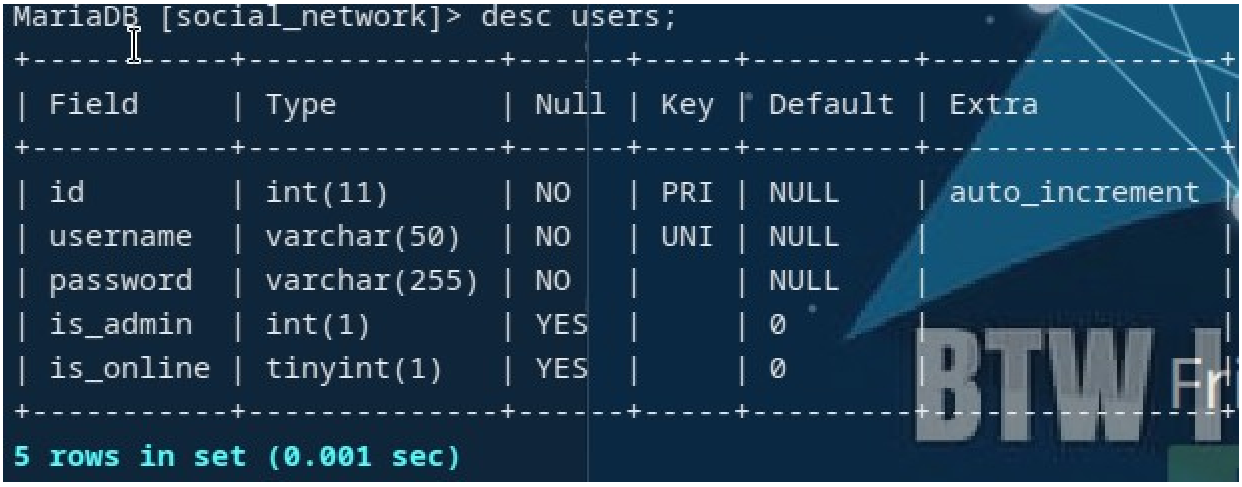
TABLE DESCRIPTION

5.1 Database Design

Users TABLE

Users Table: Users table consists of five attributes which are id ,Username, Password , is_admin and is_online. (id is used as Primary Key and Username is unique).

Desc users;



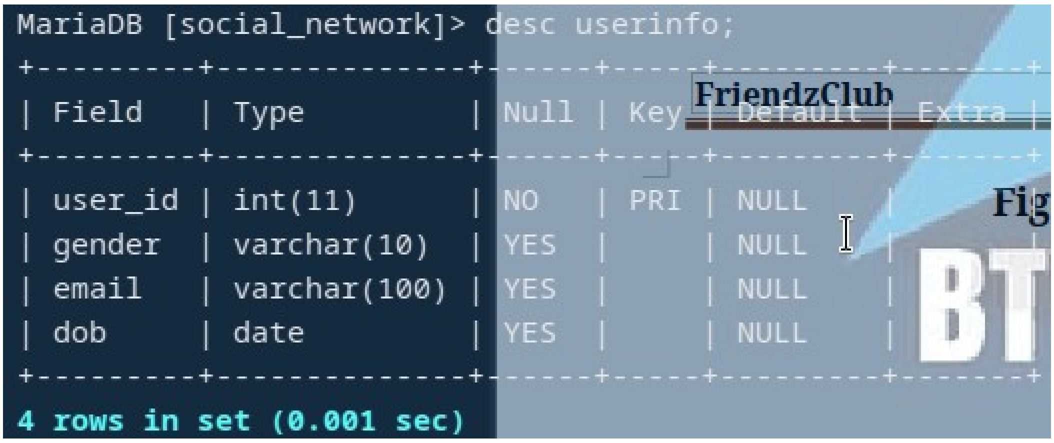
```
MariaDB [social_network]> desc users;
+-----------+--------------+------+-----+---------+----------------+
| Field     | Type         | Null | Key | Default | Extra          |
+-----------+--------------+------+-----+---------+----------------+
| id        | int(11)      | NO   | PRI | NULL    | auto_increment |
| username  | varchar(50)  | NO   | UNI | NULL    |                |
| password  | varchar(255) | NO   |     | NULL    |                |
| is_admin  | int(1)       | YES  |     | 0       |                |
| is_online | tinyint(1)   | YES  |     | 0       |                |
+-----------+--------------+------+-----+---------+----------------+
5 rows in set (0.001 sec)
```

Fig 5.1 Users table description

userinfo TABLE

userinfo Table: userinfo table consists of three attributes which are user_id ,email, dob and gender. (user_id is used as Primary Key)

Desc userinfo;



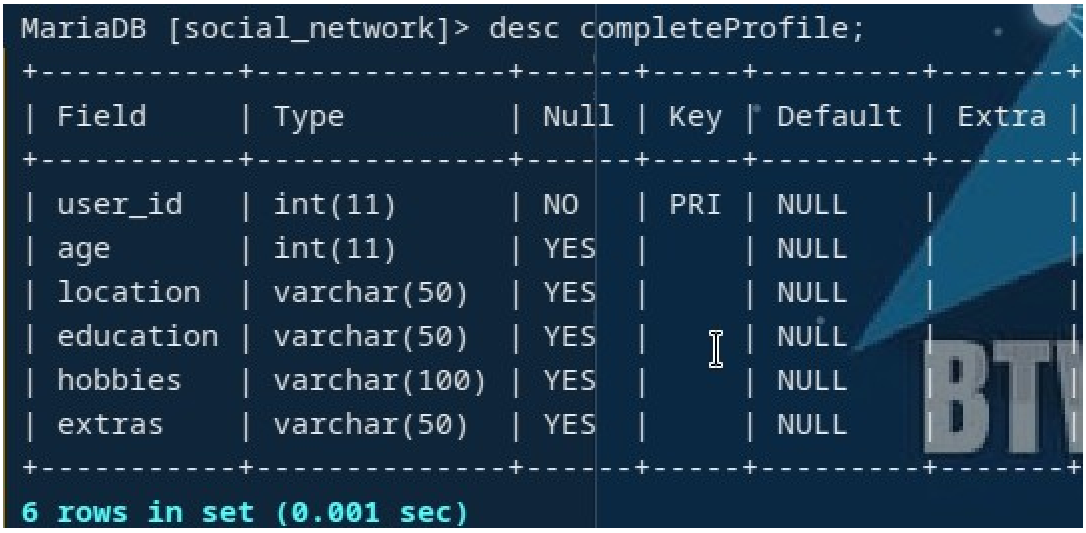
```
MariaDB [social_network]> desc userinfo;
+---------+--------------+------+-----+---------+-------+
| Field   | Type         | Null | Key | Default | Extra |
+---------+--------------+------+-----+---------+-------+
| user_id | int(11)      | NO   | PRI | NULL    |       |
| gender  | varchar(10)  | YES  |     | NULL    |       |
| email   | varchar(100) | YES  |     | NULL    |       |
| dob     | date         | YES  |     | NULL    |       |
+---------+--------------+------+-----+---------+-------+
4 rows in set (0.001 sec)
```

Fig 5.2 userinfo table description

CompleteProfile TABLE

CompleteProfile Table: Users table consists of six attributes which are user_id , age , location , education , hobbies and extras. (id is used as Primary Key).

Desc CompleteProfile;



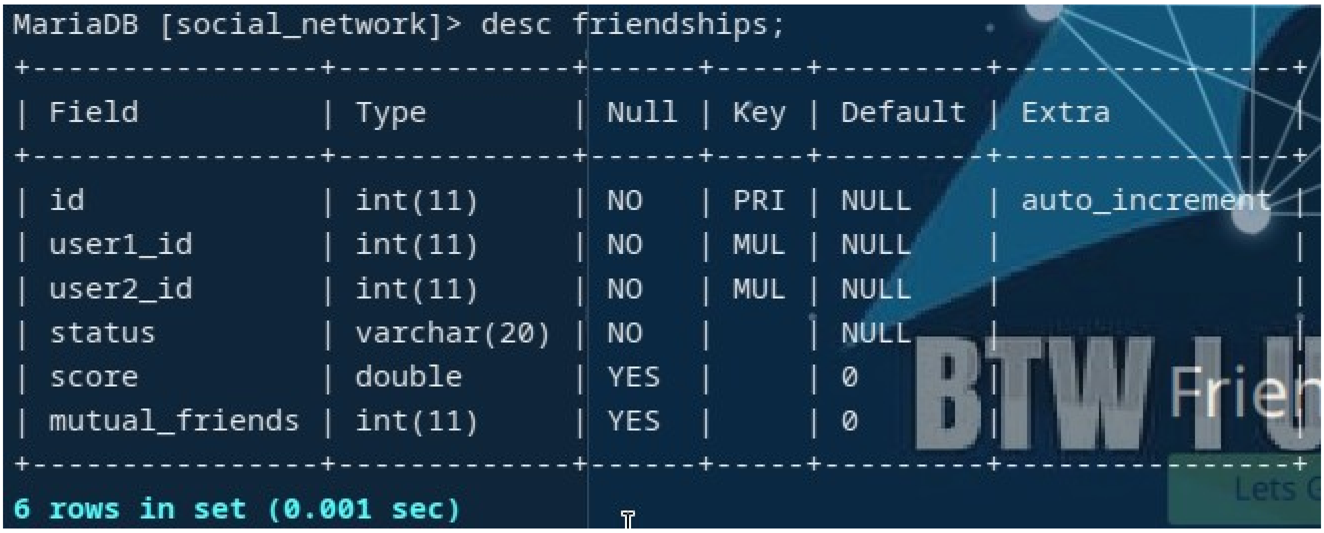
```
MariaDB [social_network]> desc completeProfile;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| user_id   | int(11)      | NO   | PRI | NULL    |       |
| age       | int(11)      | YES  |     | NULL    |       |
| location  | varchar(50)  | YES  |     | NULL    |       |
| education | varchar(50)  | YES  |     | NULL    |       |
| hobbies   | varchar(100) | YES  |     | NULL    |       |
| extras    | varchar(50)  | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
6 rows in set (0.001 sec)
```

Fig 5.3 CompleteProfile table description

friendships TABLE

friendships Table: Users table consists of six attributes which are id , user1_id , user2_id , status , score and mutual_friends. (id is used as Primary Key , user1_id and user2_id are Foreign keys references users table(id)).

Desc friendships;



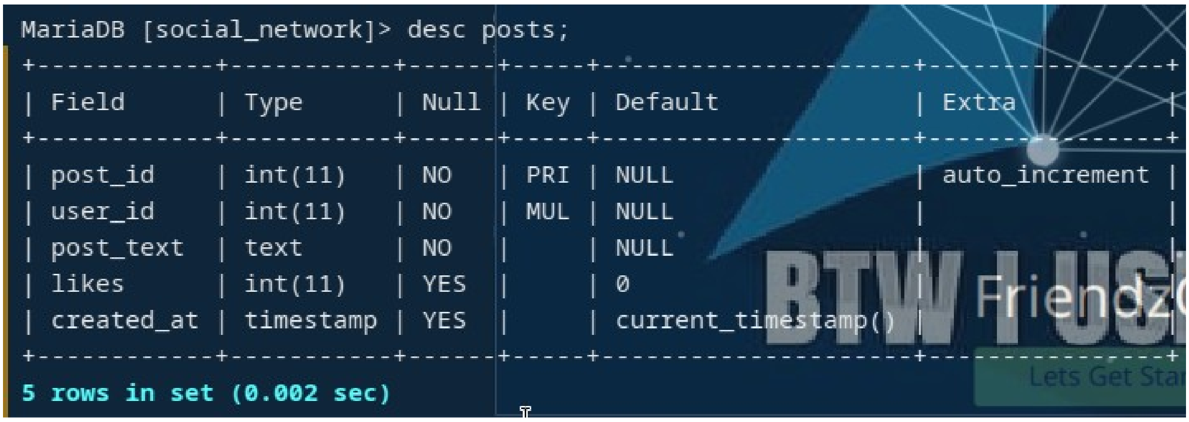
```
MariaDB [social_network]> desc friendships;
+----------------+-------------+------+-----+---------+----------------+
| Field          | Type        | Null | Key | Default | Extra          |
+----------------+-------------+------+-----+---------+----------------+
| id             | int(11)     | NO   | PRI | NULL    | auto_increment |
| user1_id       | int(11)     | NO   | MUL | NULL    |                |
| user2_id       | int(11)     | NO   | MUL | NULL    |                |
| status         | varchar(20) | NO   |     | NULL    |                |
| score          | double      | YES  |     | 0       |                |
| mutual_friends | int(11)     | YES  |     | 0       |                |
+----------------+-------------+------+-----+---------+----------------+
6 rows in set (0.001 sec)
```

Fig 5.4 friendships table description

posts TABLE

posts Table: Users table consists of five attributes which are post_id , user_id , post_text , likes , created_at. (id is used as Primary Key , user1_id and is used as Foreign keys references users table(id)).

Desc posts;



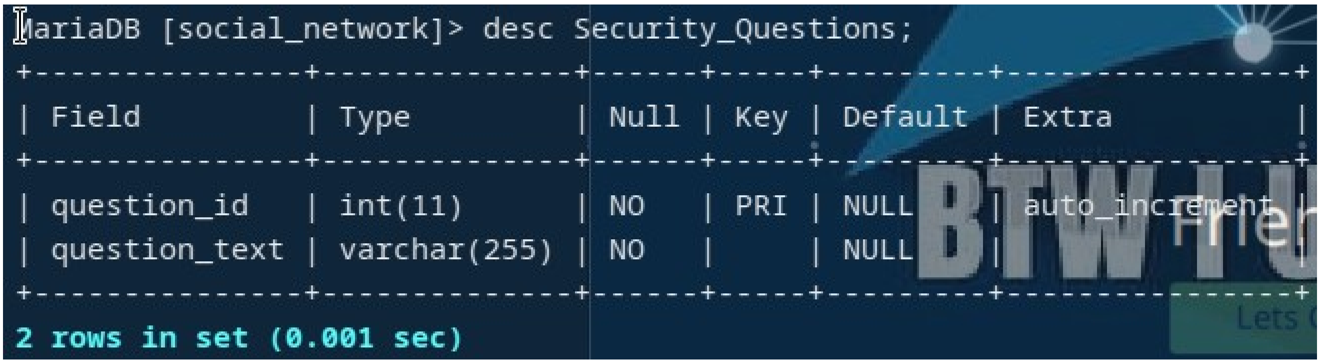
```
MariaDB [social_network]> desc posts;
+------------+-----------+------+-----+---------------------+----------------+
| Field      | Type      | Null | Key | Default             | Extra          |
+------------+-----------+------+-----+---------------------+----------------+
| post_id    | int(11)   | NO   | PRI | NULL                | auto_increment |
| user_id    | int(11)   | NO   | MUL | NULL                |                |
| post_text  | text      | NO   |     | NULL                |                |
| likes      | int(11)   | YES  |     | 0                   |                |
| created_at | timestamp | YES  |     | current_timestamp() |                |
+------------+-----------+------+-----+---------------------+----------------+
5 rows in set (0.002 sec)
```

Fig 5.5 posts table description

Security Questions TABLE

Security Questions Table: Users table consists of two attributes which are question_id , question_text . (id is used as Primary Key).

Desc Security_Questions;



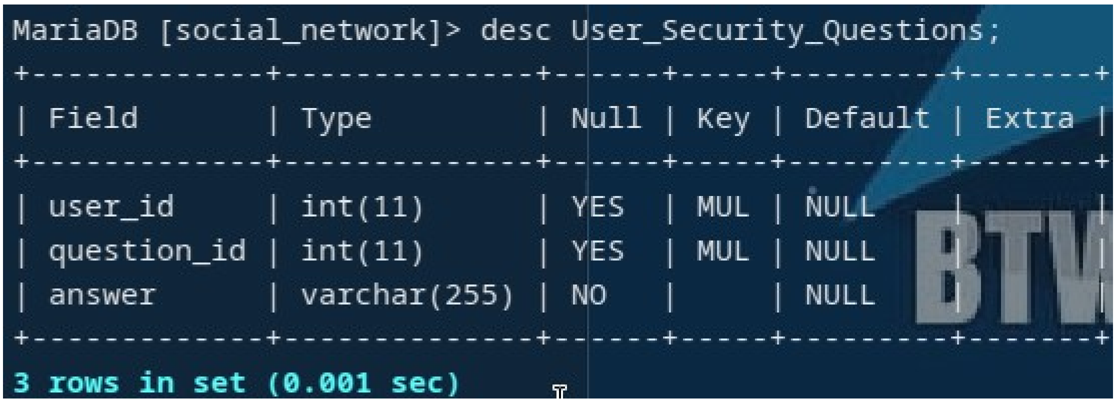
```
MariaDB [social_network]> desc Security_Questions;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| question_id   | int(11)      | NO   | PRI | NULL    | auto_increment |
| question_text | varchar(255) | NO   |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
2 rows in set (0.001 sec)
```

Fig 5.6 security questions table description

User Security Questions TABLE

User Security Questions Table: Users table consists of two attributes which are user_id , question_id , answer . (user_id is used as Foriegn Key references users(id) and question_id is used as Foreign Key references security_questions(question_id)).

Desc User_Security_Questions;



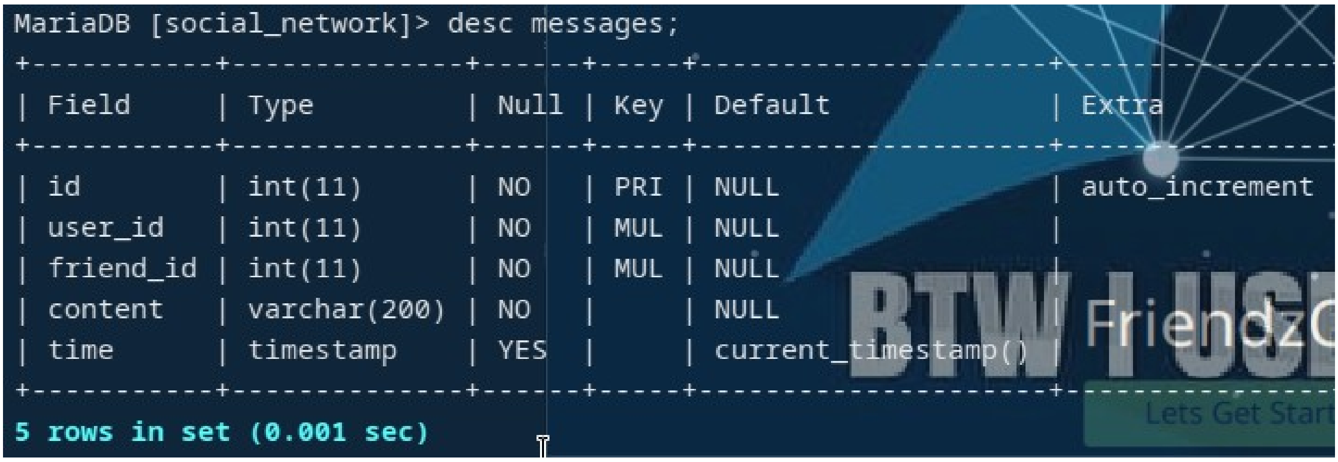
```
MariaDB [social_network]> desc User_Security_Questions;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| user_id     | int(11)      | YES  | MUL | NULL    |       |
| question_id | int(11)      | YES  | MUL | NULL    |       |
| answer      | varchar(255) | NO   |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
3 rows in set (0.001 sec)
```

Fig 5.7 user security questions table description

Messages TABLE

Messages Table: Users table consists of five attributes which are id ,user_id , friend_id , content, time. (id is used as Primary Key and used_id and friend_id is used as Foriegn Key references users(id)).

Desc messages;



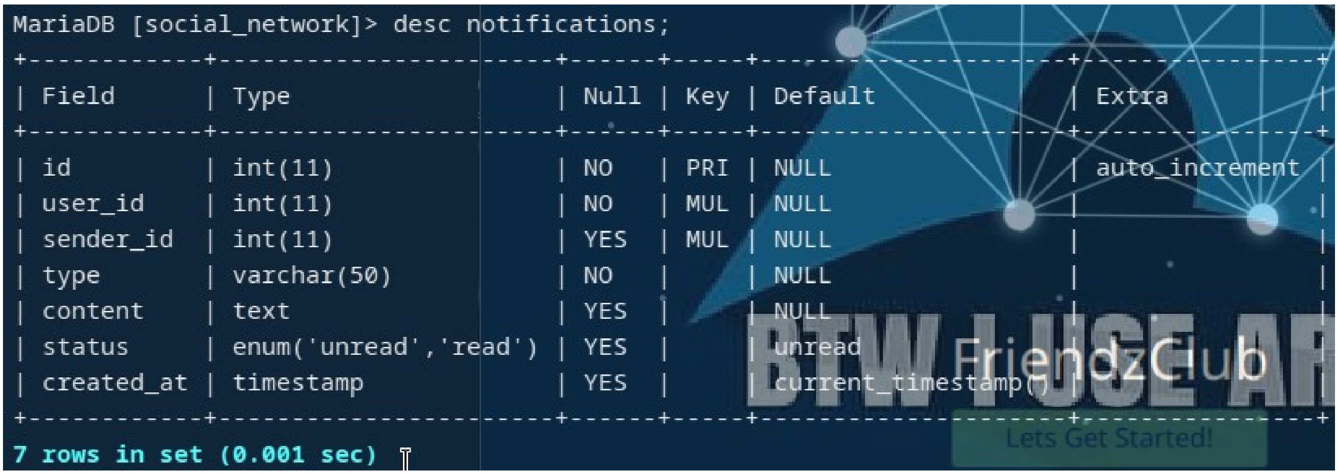
```
MariaDB [social_network]> desc messages;
+-----------+--------------+------+-----+---------------------+--------------
| Field     | Type         | Null | Key | Default             | Extra
+-----------+--------------+------+-----+---------------------+--------------
| id        | int(11)      | NO   | PRI | NULL                | auto_increment
| user_id   | int(11)      | NO   | MUL | NULL                |
| friend_id | int(11)      | NO   | MUL | NULL                |
| content   | varchar(200) | NO   |     | NULL                |
| time      | timestamp    | YES  |     | current_timestamp() |
+-----------+--------------+------+-----+---------------------+--------------
5 rows in set (0.001 sec)
```

Fig 5.8 messages table description

Notifcations TABLE

Notifcations Table: Users table consists of 7 attributes which are id ,user_id , sender_id ,type , content, status, status , created_at. (id is used as Primary Key and used_id and friend_id is used as Foriegn Key references users(id)).

Desc notifications;



```
MariaDB [social_network]> desc notifications;
+------------+----------------------+------+-----+-------------------+----------------+
| Field      | Type                 | Null | Key | Default           | Extra          |
+------------+----------------------+------+-----+-------------------+----------------+
| id         | int(11)              | NO   | PRI | NULL              | auto_increment |
| user_id    | int(11)              | NO   | MUL | NULL              |                |
| sender_id  | int(11)              | YES  | MUL | NULL              |                |
| type       | varchar(50)          | NO   |     | NULL              |                |
| content    | text                 | YES  |     | NULL              |                |
| status     | enum('unread','read') | YES  |     | unread            |                |
| created_at | timestamp            | YES  |     | current_timestamp() |              |
+------------+----------------------+------+-----+-------------------+----------------+
7 rows in set (0.001 sec)
```

Fig 5.9 notifications table description

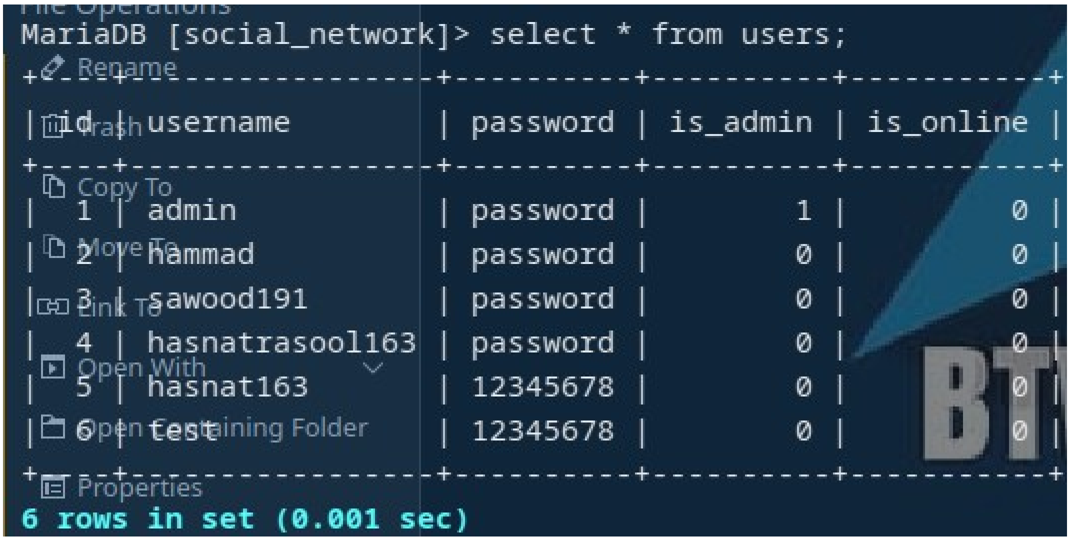
CHAPTER 6

TABLE WITH VALUES

6.1 Output design:

users table : Users table consists of five attributes which are used in authentication , registeration .

Select * from users;

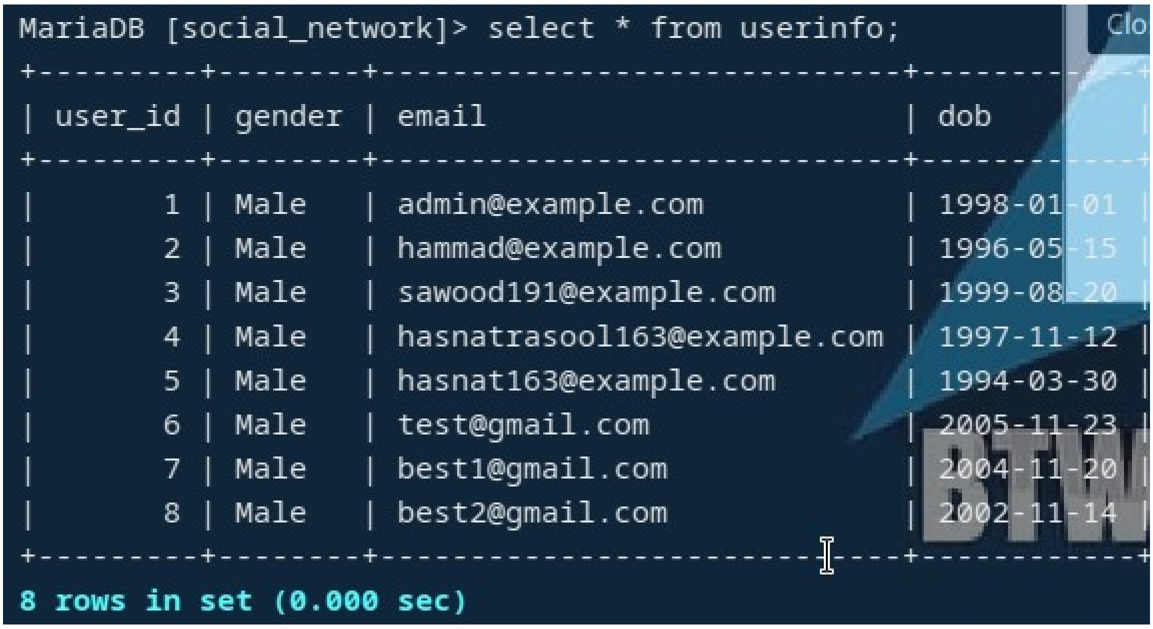


```
MariaDB [social_network]> select * from users;
+----+-----------------+----------+----------+-----------+
| id | username        | password | is_admin | is_online |
+----+-----------------+----------+----------+-----------+
|  1 | admin           | password |        1 |         0 |
|  2 | hammad          | password |        0 |         0 |
|  3 | sawood191       | password |        0 |         0 |
|  4 | hasnatrasool163 | password |        0 |         0 |
|  5 | hasnat163       | 12345678 |        0 |         0 |
|  6 | test            | 12345678 |        0 |         0 |
+----+-----------------+----------+----------+-----------+
6 rows in set (0.001 sec)
```

Table 6.1 users table

userinfo table :userinfo table consists of four attributes which are used in recommendations.

Select * from userinfo;

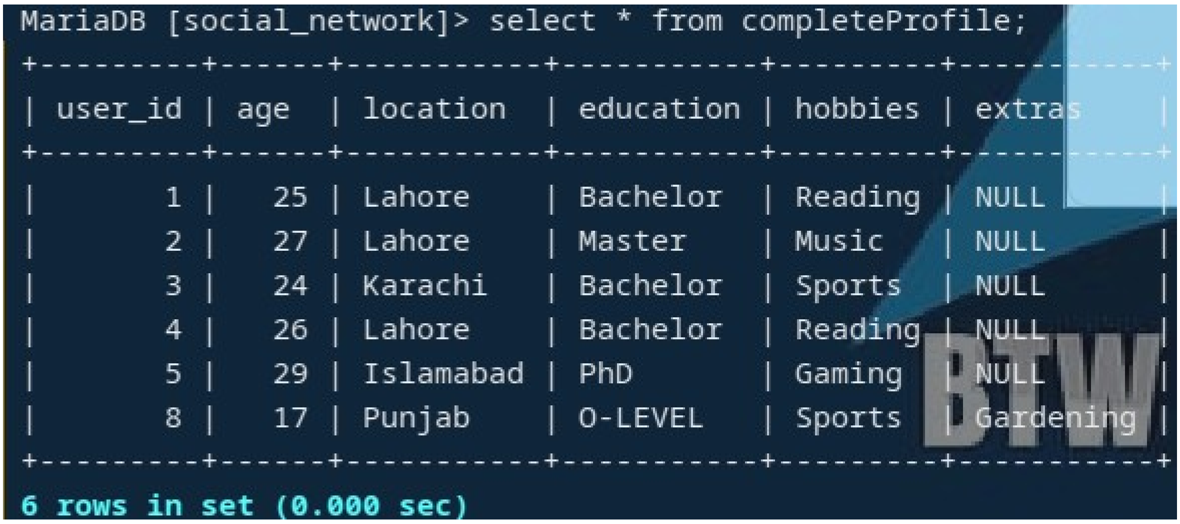


```
MariaDB [social_network]> select * from userinfo;
+---------+--------+----------------------------+------------+
| user_id | gender | email                      | dob        |
+---------+--------+----------------------------+------------+
|       1 | Male   | admin@example.com          | 1998-01-01 |
|       2 | Male   | hammad@example.com         | 1996-05-15 |
|       3 | Male   | sawood191@example.com      | 1999-08-20 |
|       4 | Male   | hasnatrasool163@example.com | 1997-11-12 |
|       5 | Male   | hasnat163@example.com      | 1994-03-30 |
|       6 | Male   | test@gmail.com             | 2005-11-23 |
|       7 | Male   | best1@gmail.com            | 2004-11-20 |
|       8 | Male   | best2@gmail.com            | 2002-11-14 |
+---------+--------+----------------------------+------------+
8 rows in set (0.000 sec)
```

Table 6.2 users table

completeProfile table :completeProfile table consists of six attributes which are used in recommendations for users,

Select * from completeProfile;

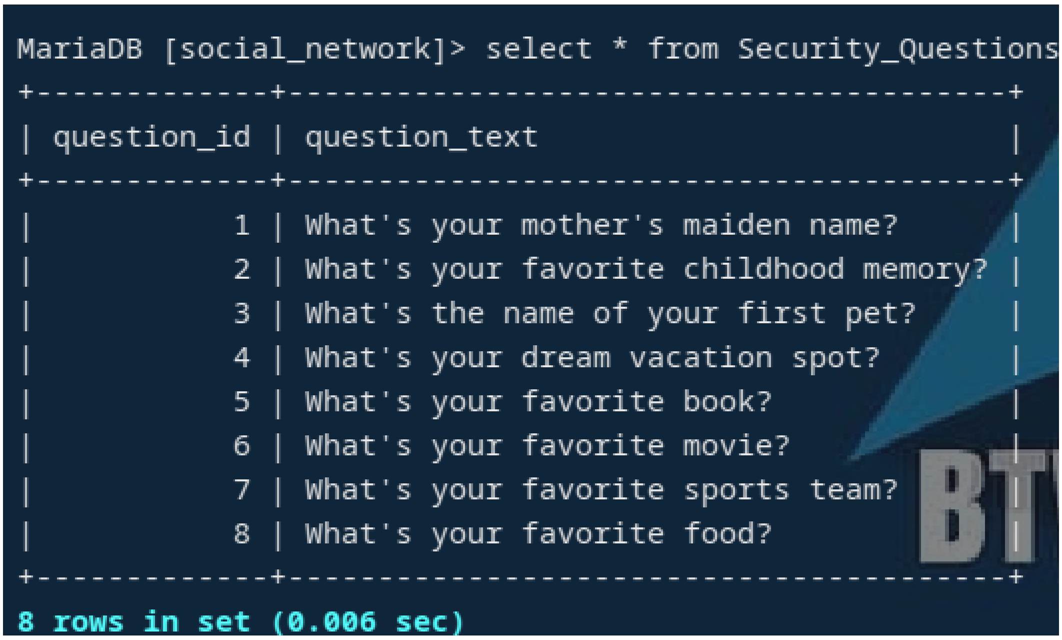


```
MariaDB [social_network]> select * from completeProfile;
+---------+-----+-----------+-----------+---------+-----------+
| user_id | age | location  | education | hobbies | extras    |
+---------+-----+-----------+-----------+---------+-----------+
|       1 |  25 | Lahore    | Bachelor  | Reading | NULL      |
|       2 |  27 | Lahore    | Master    | Music   | NULL      |
|       3 |  24 | Karachi   | Bachelor  | Sports  | NULL      |
|       4 |  26 | Lahore    | Bachelor  | Reading | NULL      |
|       5 |  29 | Islamabad | PhD       | Gaming  | NULL      |
|       8 |  17 | Punjab    | O-LEVEL   | Sports  | Gardening |
+---------+-----+-----------+-----------+---------+-----------+
6 rows in set (0.000 sec)
```

Table 6.3 users table

Security questions table :seurity questions table consists of two attributes which are used to show default questions to user for security .

Select * from Security_Questions;

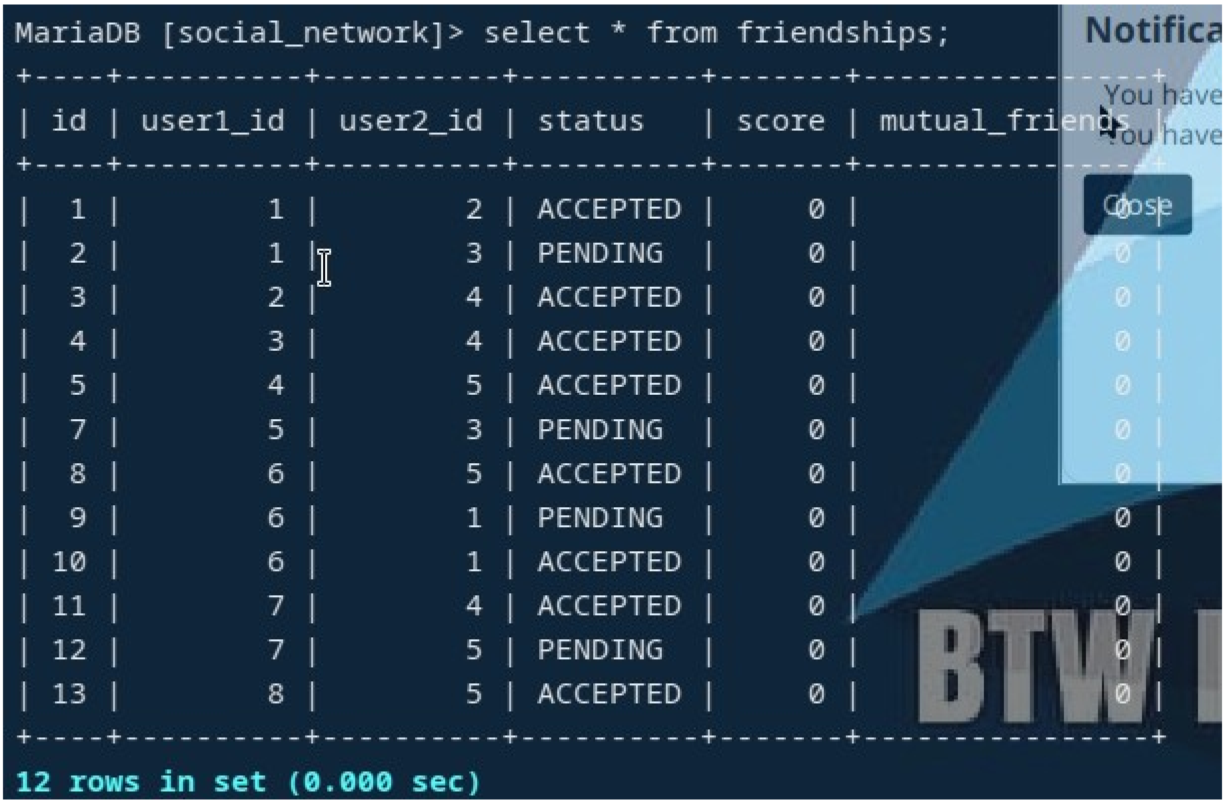


```
MariaDB [social_network]> select * from Security_Questions
+-------------+-----------------------------------------+
| question_id | question_text                           |
+-------------+-----------------------------------------+
|           1 | What's your mother's maiden name?       |
|           2 | What's your favorite childhood memory?  |
|           3 | What's the name of your first pet?      |
|           4 | What's your dream vacation spot?        |
|           5 | What's your favorite book?              |
|           6 | What's your favorite movie?             |
|           7 | What's your favorite sports team?       |
|           8 | What's your favorite food?              |
+-------------+-----------------------------------------+
8 rows in set (0.006 sec)
```

Table 6.4 Security_Questions table

friendships table :frendships table consists of six attributes which are used to store users relations and friend requests status.

Select * from friendships;

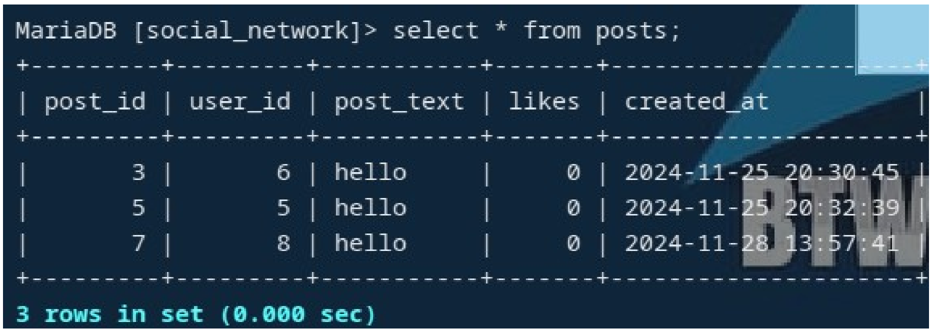


```
MariaDB [social_network]> select * from friendships;
+----+----------+----------+----------+-------+----------------+
| id | user1_id | user2_id | status   | score | mutual_friends |
+----+----------+----------+----------+-------+----------------+
|  1 |        1 |        2 | ACCEPTED |     0 |              0 |
|  2 |        1 |        3 | PENDING  |     0 |              0 |
|  3 |        2 |        4 | ACCEPTED |     0 |              0 |
|  4 |        3 |        4 | ACCEPTED |     0 |              0 |
|  5 |        4 |        5 | ACCEPTED |     0 |              0 |
|  7 |        5 |        3 | PENDING  |     0 |              0 |
|  8 |        6 |        5 | ACCEPTED |     0 |              0 |
|  9 |        6 |        1 | PENDING  |     0 |              0 |
| 10 |        6 |        1 | ACCEPTED |     0 |              0 |
| 11 |        7 |        4 | ACCEPTED |     0 |              0 |
| 12 |        7 |        5 | PENDING  |     0 |              0 |
| 13 |        8 |        5 | ACCEPTED |     0 |              0 |
+----+----------+----------+----------+-------+----------------+
12 rows in set (0.000 sec)
```

Table 6.5 Security_Questions table

posts table : Users table consists of five attributes which are used to store and show posts of users.

Select * from posts;

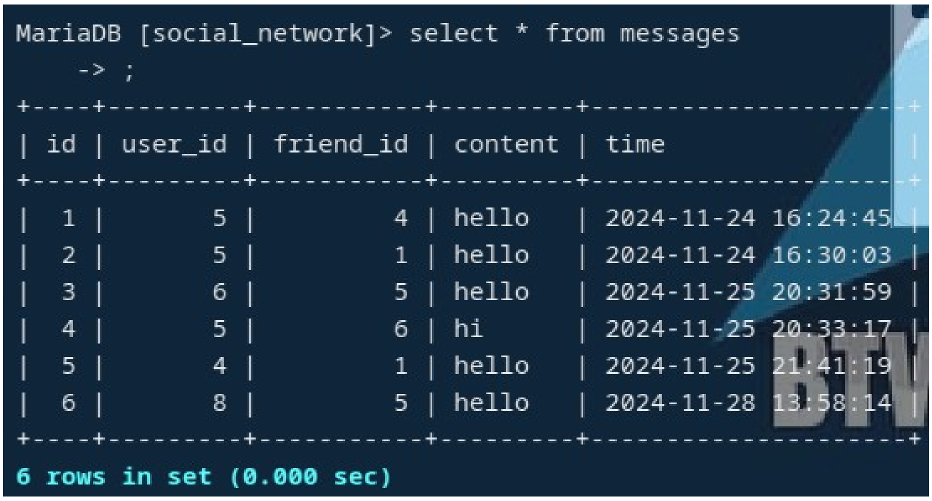


```
MariaDB [social_network]> select * from posts;
+---------+---------+-----------+-------+---------------------+
| post_id | user_id | post_text | likes | created_at          |
+---------+---------+-----------+-------+---------------------+
|       3 |       6 | hello     |     0 | 2024-11-25 20:30:45 |
|       5 |       5 | hello     |     0 | 2024-11-25 20:32:39 |
|       7 |       8 | hello     |     0 | 2024-11-28 13:57:41 |
+---------+---------+-----------+-------+---------------------+
3 rows in set (0.000 sec)
```

Table 6.6 posts table

messages table :messages table consists of five attributes which are used to store and show message to users .

Select * from messages;



```
MariaDB [social_network]> select * from messages
    -> ;
+----+---------+-----------+---------+---------------------+
| id | user_id | friend_id | content | time                |
+----+---------+-----------+---------+---------------------+
|  1 |       5 |         4 | hello   | 2024-11-24 16:24:45 |
|  2 |       5 |         1 | hello   | 2024-11-24 16:30:03 |
|  3 |       6 |         5 | hello   | 2024-11-25 20:31:59 |
|  4 |       5 |         6 | hi      | 2024-11-25 20:33:17 |
|  5 |       4 |         1 | hello   | 2024-11-25 21:41:19 |
|  6 |       8 |         5 | hello   | 2024-11-28 13:58:14 |
+----+---------+-----------+---------+---------------------+
6 rows in set (0.000 sec)
```

Table 6.7 messages table

CHAPTER 7

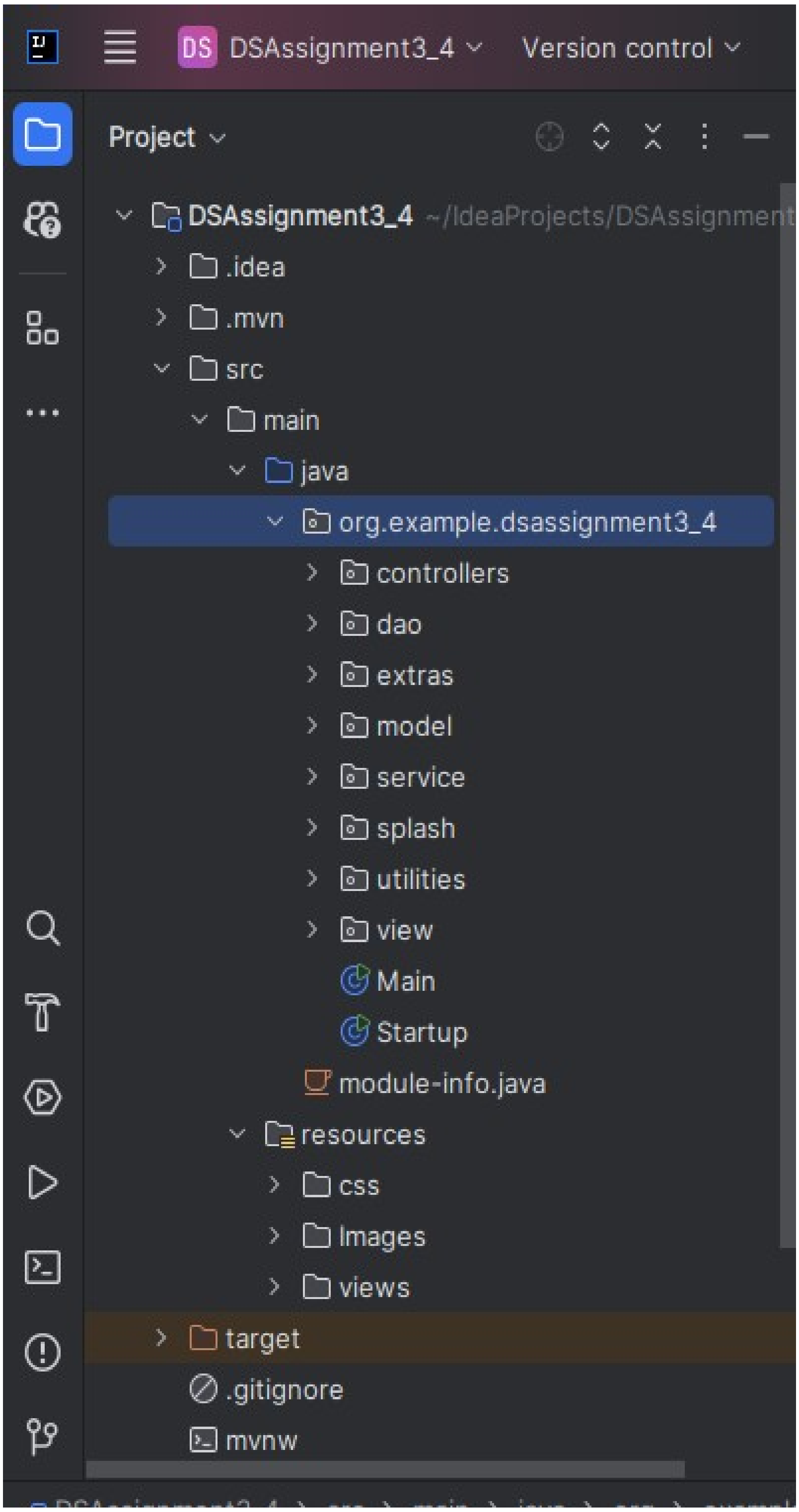
7.1Implementation

Sample code :

```
package org.example.dsassignment3_4;

import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Parent;
import javafx.scene.Scene;
import javafx.stage.Stage;

import java.util.Objects;


public class Main extends Application {

   public static void main(String[] args) {
      launch(args);
   }

   @Override
   public void start(Stage stage) throws Exception {
      try {
         FXMLLoader loader = new
FXMLLoader(Objects.requireNonNull(getClass().getResource("/views/splash.fxml")));
         Parent root = loader.load();
         Scene scene = new Scene(root);
         scene.getStylesheets().add(String.valueOf(getClass().getResource("/css/
GlobalStyles.css")));
         stage.setTitle("Friend Connections Graph");
         stage.setScene(scene);
         stage.centerOnScreen();
         stage.setResizable(false);
         stage.show();
      } catch (Exception e) {
         e.printStackTrace();
      }
   }
```

}

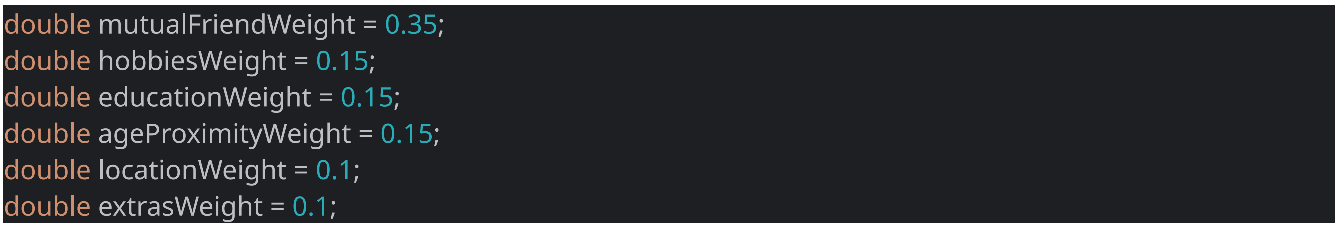
7.2Packages Structure:



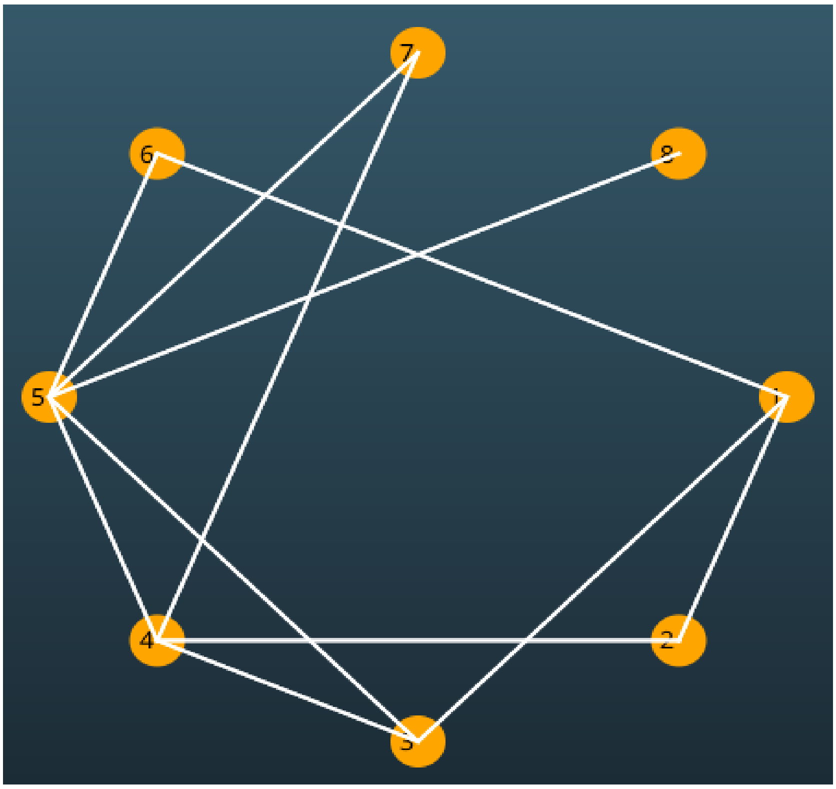
```
DSAssignment3_4 ~/IdeaProjects/DSAssignment
    .idea
    .mvn
    src
        main
            java
                org.example.dsassignment3_4
                    controllers
                    dao
                    extras
                    model
                    service
                    splash
                    utilities
                    view
                    Main
                    Startup
                module-info.java
            resources
                css
                Images
                views
    target
    .gitignore
    mvnw
```

7.3Friend -Recommendation- Formula :

Weights :



```
double mutualFriendWeight = 0.35;
double hobbiesWeight = 0.15;
double educationWeight = 0.15;
double ageProximityWeight = 0.15;
double locationWeight = 0.1;
double extrasWeight = 0.1;
```



e.g graph representation :
{

1 = [2, 3, 6],

2 = [1, 4],

3 = [1 , 4, 5],

4 = [2 , 3, 5, 7],

5 = [3, 4, 6 , 7 , 8]

6= [1 , 5],

7 = [4 ,5]

8 = [5]

}

Total Users : 8

Find friend suggestion for user (4):
Get direct friends: [2 , 3 ,5 ,7]

Get friends of friends

(excluding 4 itself)

2 = [1]

3 = [1, 5]

5= [3 , 6 , 7 ,8]

7= [5]

Approach :

1) use bfs to find friends of friend [1 ,6 ,8] all new unique possible suggestions

1 and 8 share some mutual friends.

7.4Final Formula :

MutualFriendsW * (mutual friends /total friends) + HobbiesW + LocationW + AgeW + EducationW + ExtrasW .

Max score 1.

Total Levels 3

- 0.8 – 1.0 High
- 0.5 – 0.8 Medium
- 0.0 – 0.49 Low

Get Mutual friends : user4 share 2 mutual users with user1. [2,3]

Get Mutual friends : user4 share 1 mutual user with user8. [5]

User 1 score = 0.35 * (2/4) + 0.15 + 0.15 +0.15 + 0.1 = 0.735 (out of 1)

User 8 score = 0.35 * (1/4) + 0 = 0.0875 (out of 1)

CHAPTER 8

TESTING

System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently before live operation commences. Testing is the process of executing the program with the intent of finding errors and missing operations and also a complete verification to determine whether the objectives are met and the user requirements are satisfied. The ultimate aim is quality assurance.

8.1 Unit Testing

The software units in a system are modules and routines that are assembled and integrated to perform a specific function. Unit testing focuses first on modules, independently of one another, to locate errors. This enables, to detect errors in coding and logic that are contained within each module. This testing includes entering data and ascertaining if the value matches to the type and size supported by java. The various controls are tested to ensure that each performs its action as required.

8.2 Integration Testing

Data can be lost across any interface, one module can have an adverse effect on another, sub functions when combined, may not produce the desired major functions. Integration testing is a systematic testing to discover errors associated within the interface. The objective is to take unit tested modules and build a program structure. All the modules are combined and tested as a whole. Here the backend module and frontend module options are integrated and tested. This testing provides the assurance that the application is well integrated functional unit with smooth transition of data.

8.3 User Acceptance

Testing User acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the system users at time of developing and making changes whenever required.

8.4 Test Cases:

Table 1.2 Test cases

Test No.	Test Name	input	Actual output	Expected output	Status
1	Login	Username and password	User is successfully Authenticated	User is successfully Authenticated	Pass
2	Login	Wrong username and password	Invalid username or password	Invalid username or password	Pass
3	Signup	User details and passsword	Account successfully created	Account successfully created	Pass
4	AddUser	Details of the user.	user inserted successfully	user inserted successfully	Pass
5	Admin Login	Username and password	Admin is successfully Authenticated	Admin is successfully Authenticated	Pass
6	FriendRequest	Send friend requets to a user	Friend request sucessfully sent	Friend request sucessfully sent	Pass
7	Profile	Open profile	View profile stats	View profile stats	Pass
8	Message	Send message to a friend or user	Message send successfully	Message send successfully	Pass
9	Notifications	open notifications page	All Notifications shown	All Notifications shown	Pass
10	Exit	Click on Exit	Logout successfully	Logout successfully	Pass

CHAPTER 9

SNAPSHOTS

1.splash screen: This scene represents the splash screen startup for 7 seconds then get started button is enabled.

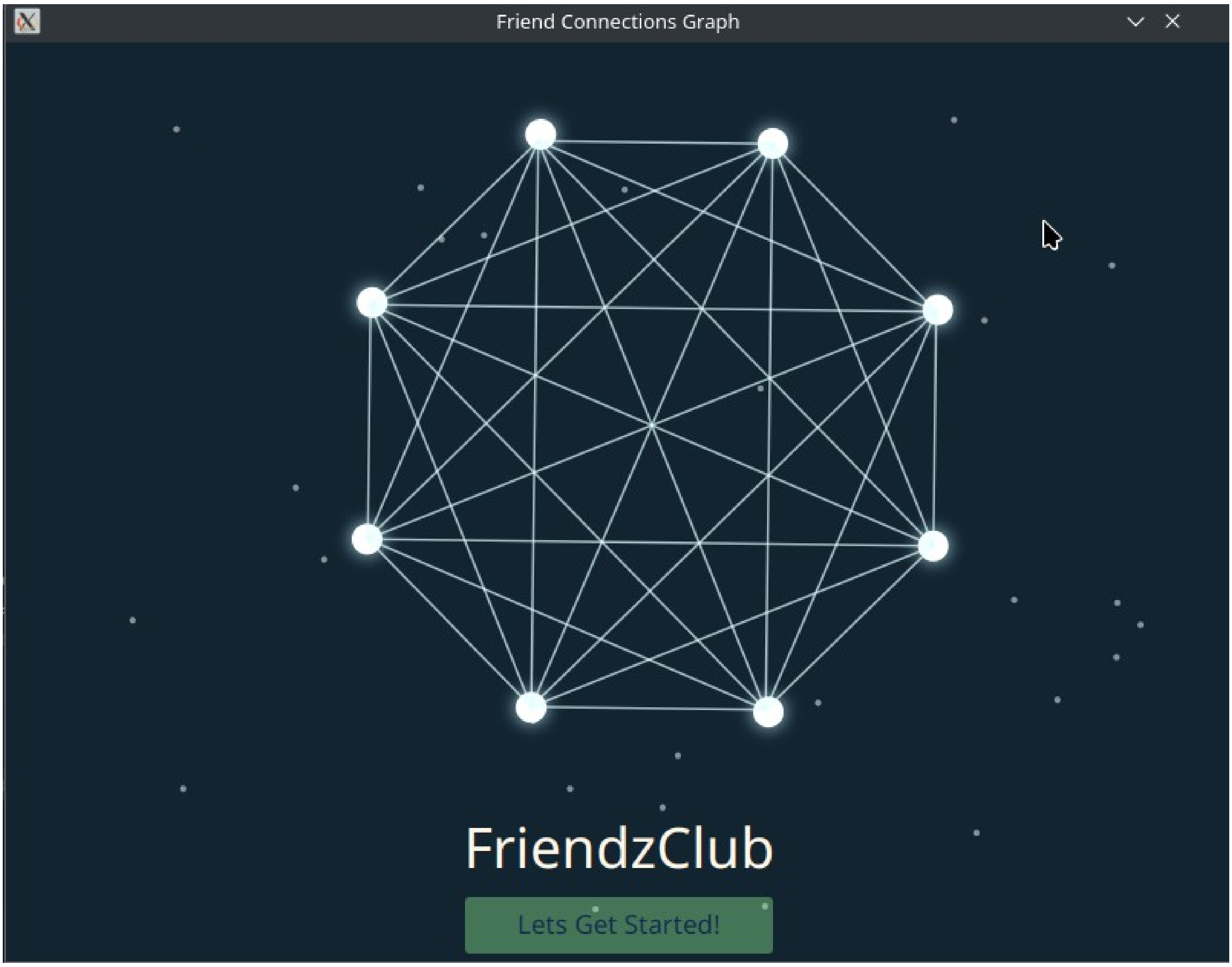
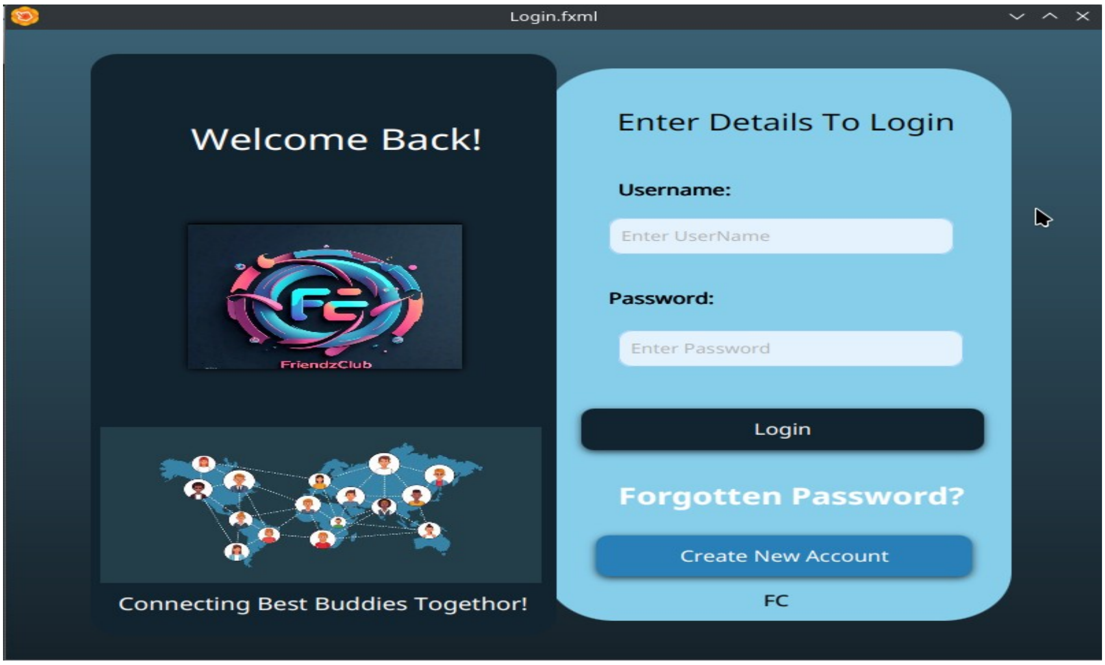


Figure 9.1: Splash scene

2.Login screen: This scene represents the login page where users can enter their details to login the system , user can move to register page or reset password from this page.



Welcome Back!

Connecting Best Buddies Togethor!

Enter Details To Login

Username:

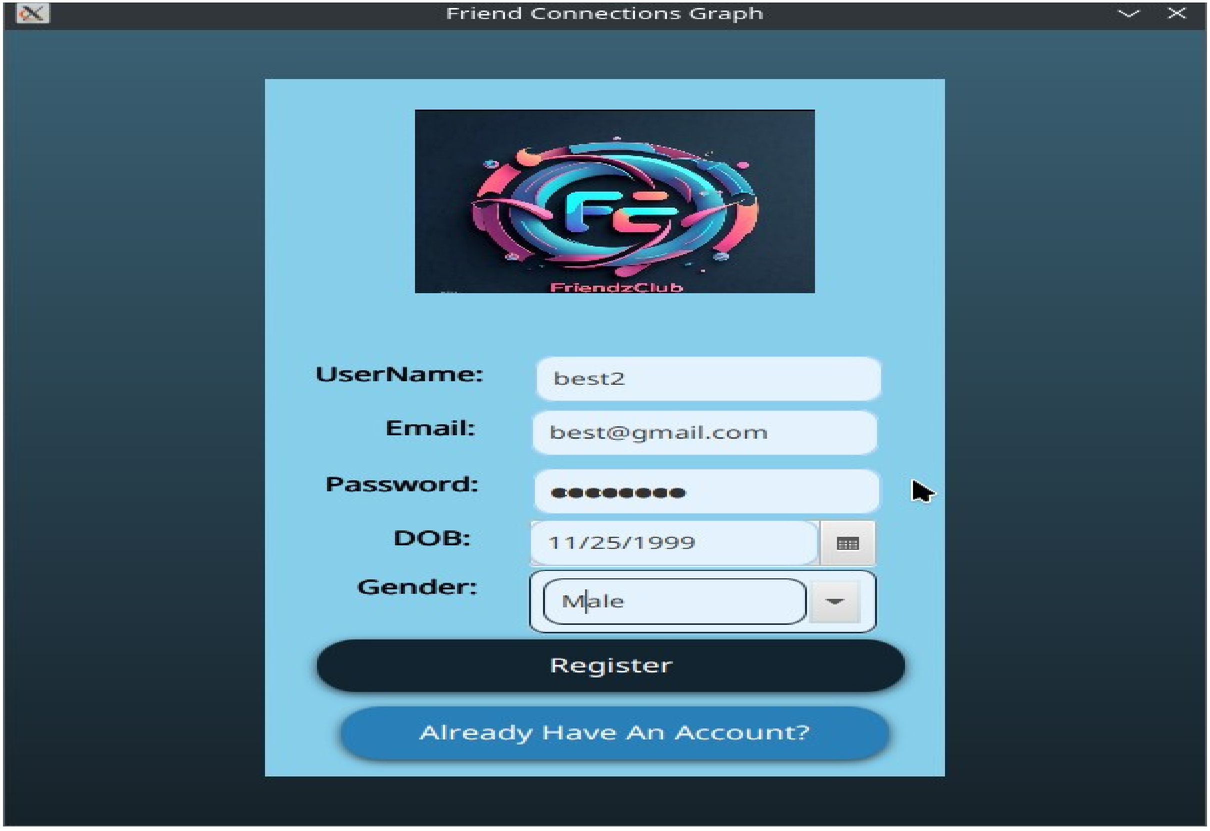
Enter UserName

Password:

Enter Password

Figure 9.2: Login scene

3.Register screen: This scene represents the register page where users can enter their details to signup in the system , user can move to login page from this page.



UserName: best2

Email: best@gmail.com

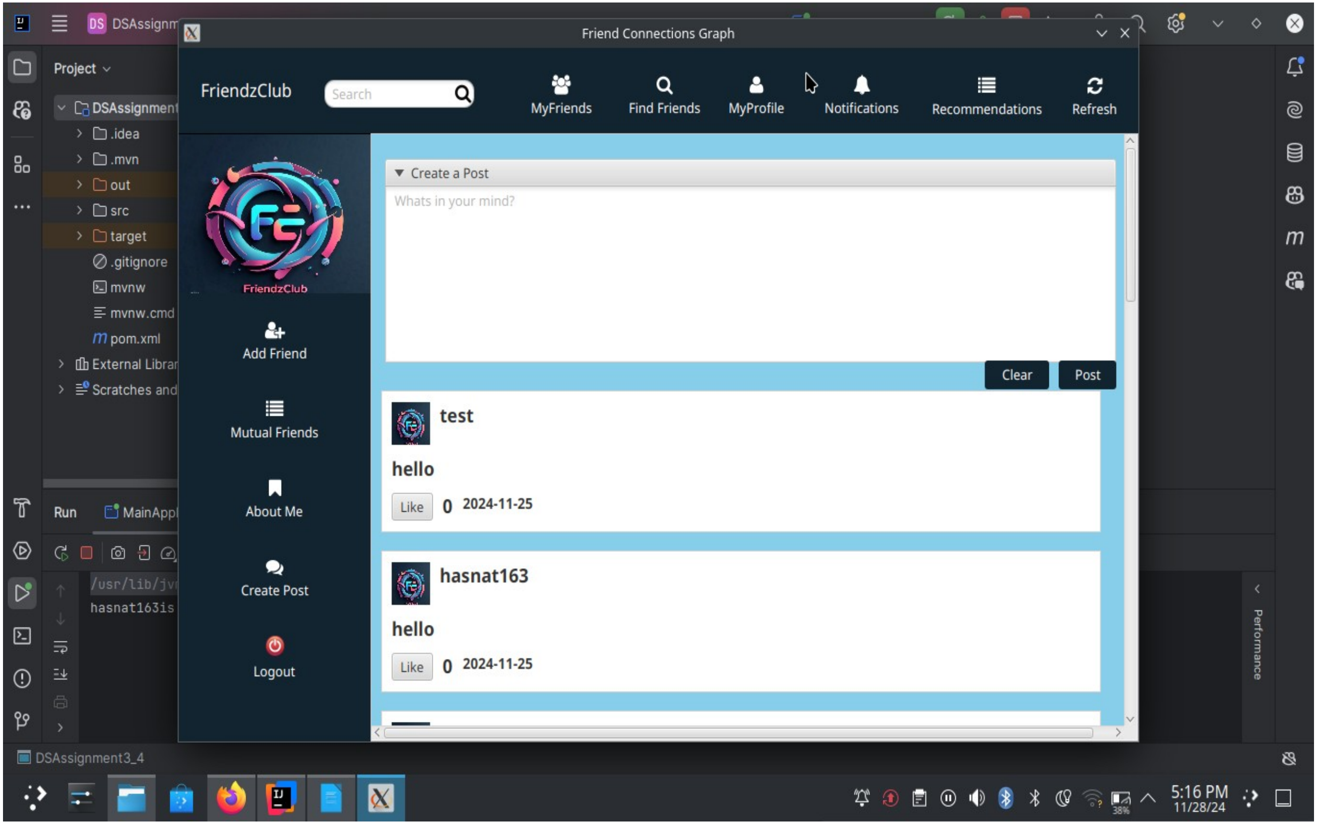
Password: ●●●●●●●●●

DOB: 11/25/1999

Gender: Male

Figure 9.3: Register scene

4.Dashboard screen: This scene represents the navigation to difference pages and shows lastest posts of users in creation order.



test

hello

Like 0 2024-11-25

hasnat163

hello

Like 0 2024-11-25

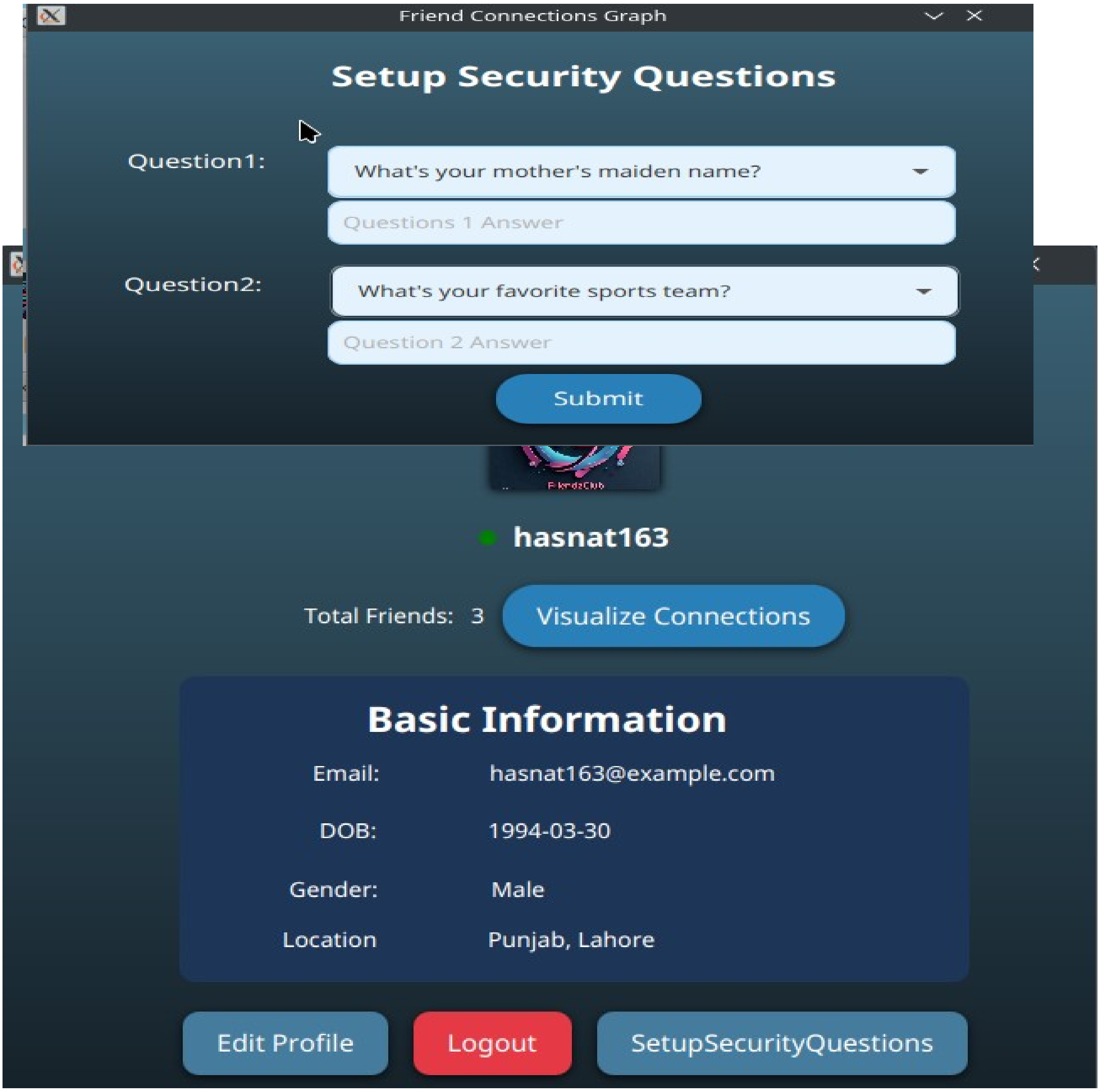
```
/usr/lib/jv
hasnat163is
```

Figure 9.4: Dashboard scene

5.Profile screen: This scene represents the profile page where users can view their basic info , navigate to edit profile , setup security questions and visualize network.

Figure 9.5: Profile scene

6.Security Questions screen: This scene represents setup security questions where user can choose questions and submit their answers..



Friend Connections Graph

Setup Security Questions

Question1: What's your mother's maiden name?

Questions 1 Answer

Question2: What's your favorite sports team?

Question 2 Answer

hasnat163

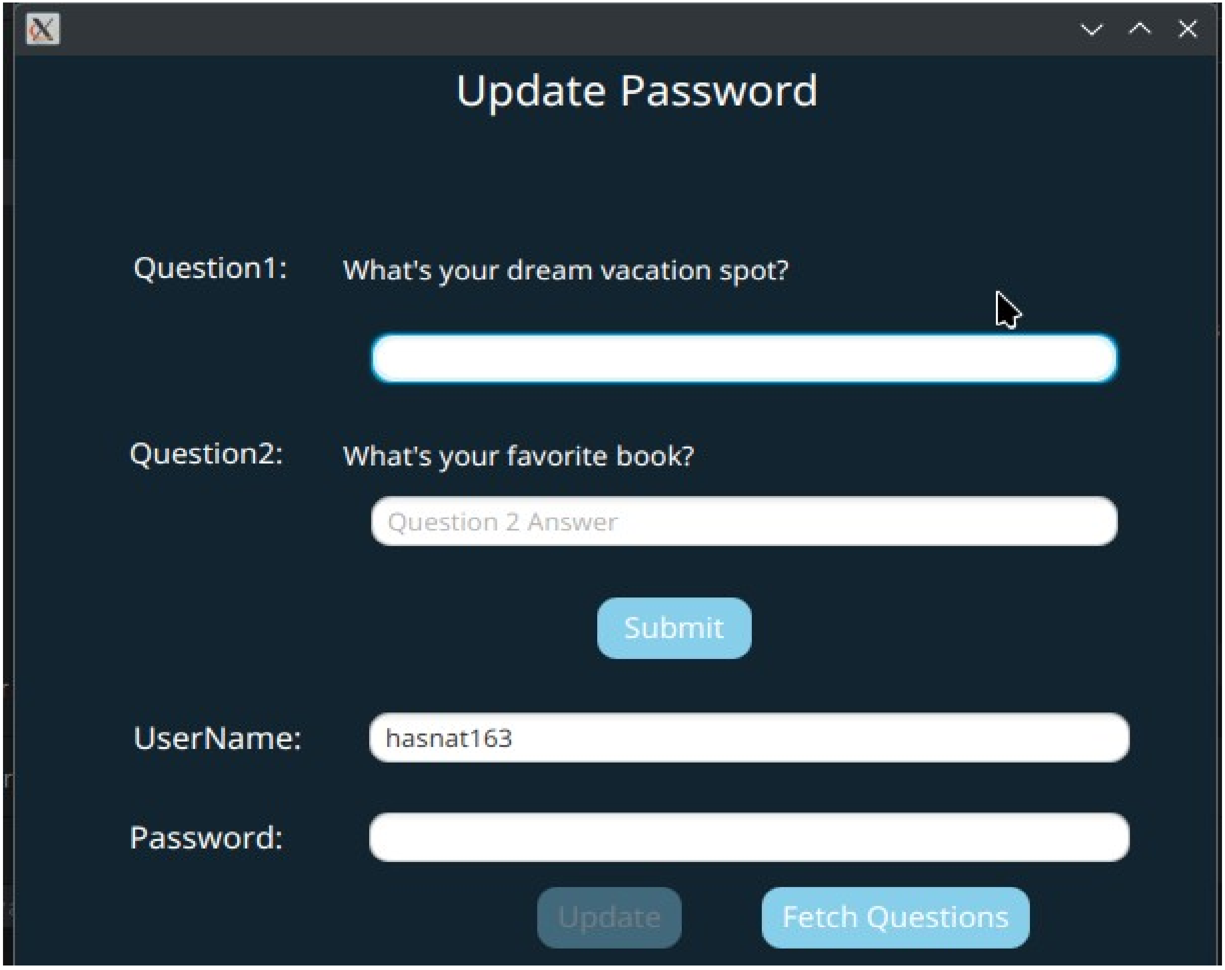
Total Friends: 3

Basic Information

Email:	hasnat163@example.com
DOB:	1994-03-30
Gender:	Male
Location	Punjab, Lahore

Figure 9.6: Security_Questions scene

7.Update Password screen: This scene represents the forget password page where user can fetch their questions and submit correct answers for recovery .



Update Password

Question1: What's your dream vacation spot?

Question2: What's your favorite book?

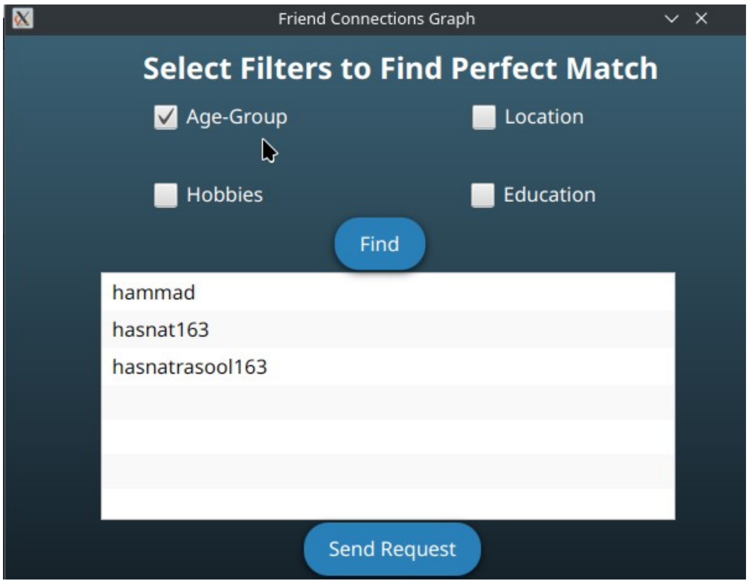
Question 2 Answer

UserName: hasnat163

Password:

Figure 9.7: Update Password scene

8.Find Friends screen: This scene represents the find friends page where user can check different combo boxes and filter to find friends and send request to them.



Friend Connections Graph

Select Filters to Find Perfect Match

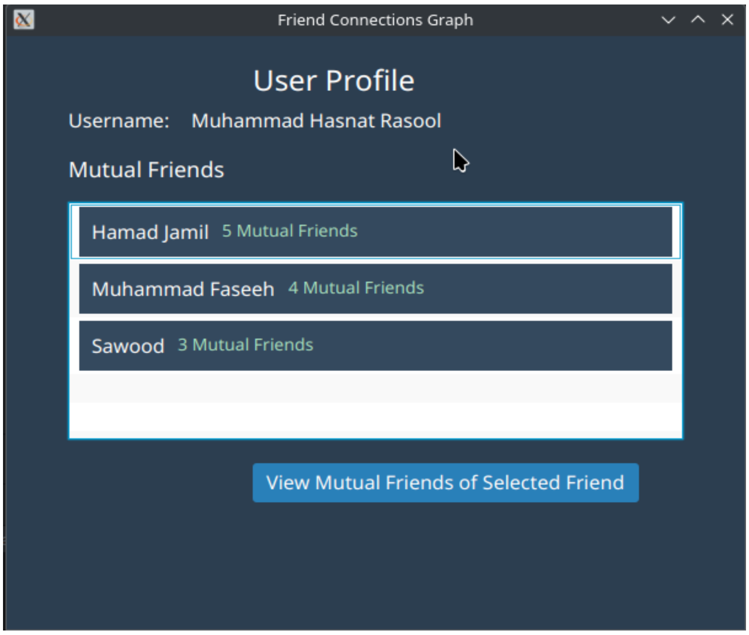
☑ Age-Group □ Location

□ Hobbies □ Education

hammad
hasnat163
hasnatrasool163

Figure 9.8: Find Friendsscene

9.Mutual Friends screen: This scene represents the mutual friends page where users can view their mutual friends with different friends .



Username: Muhammad Hasnat Rasool

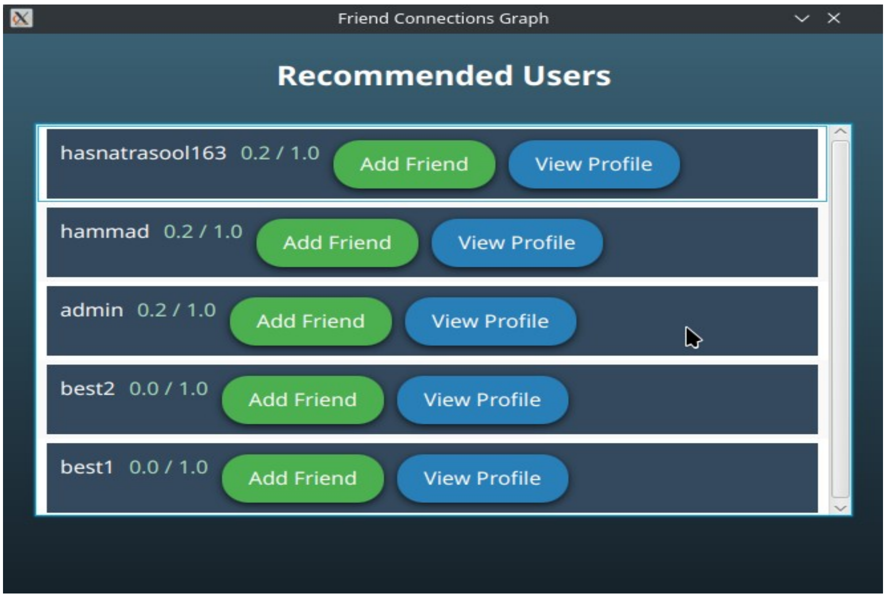
Hamad Jamil 5 Mutual Friends

Muhammad Faseeh 4 Mutual Friends

Sawood 3 Mutual Friends

Figure 9.9: Mututal Friend scene

10.Recommended Users screen: This scene represents the recommended users page where user can view top 5 recommended users with their score / level of recommendation .



hasnatrasool163 0.2 / 1.0

hammad 0.2 / 1.0

admin 0.2 / 1.0

best2 0.0 / 1.0

best1 0.0 / 1.0

Figure 9.10: Mututal Friend scene

11.Complete Profile screen: This scene represents the completeProfile page where user can complete their profile , it will be used for recommendations.

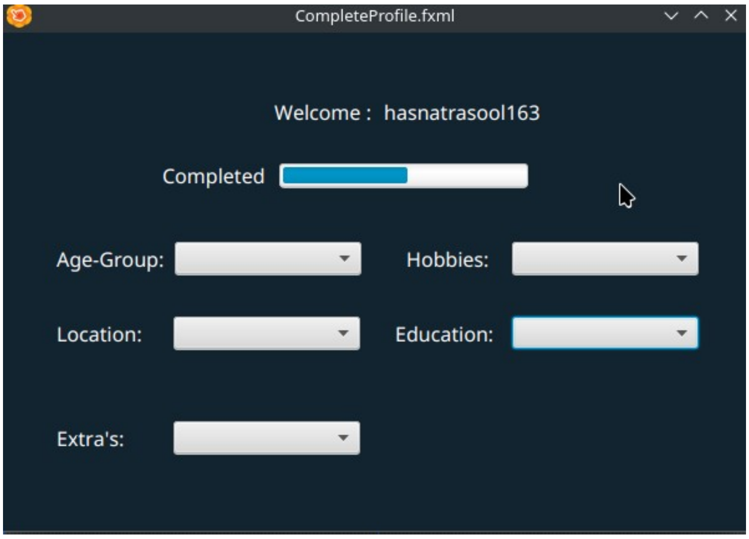


Figure 9.11: Complete Profile scene

12.MyFriends screen: This scene represents the my friends page where user can see all his friends and navigate to their profile or unfriend them .

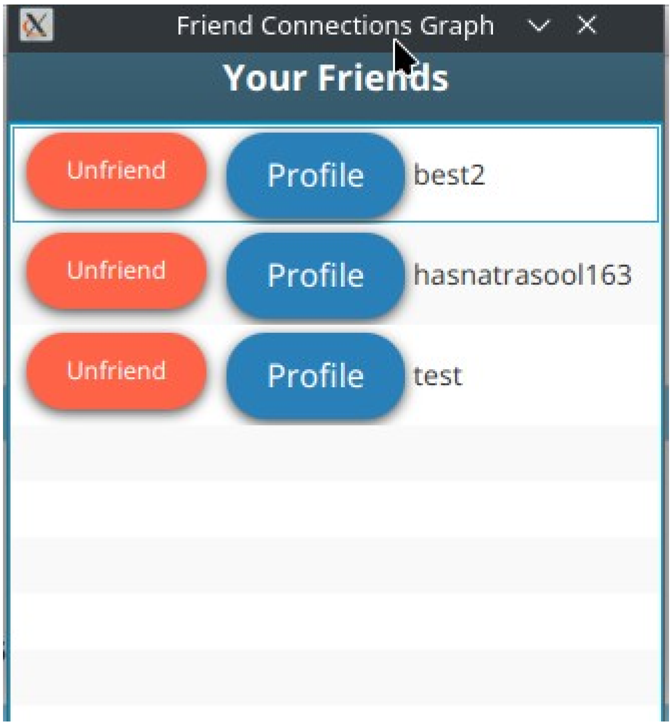
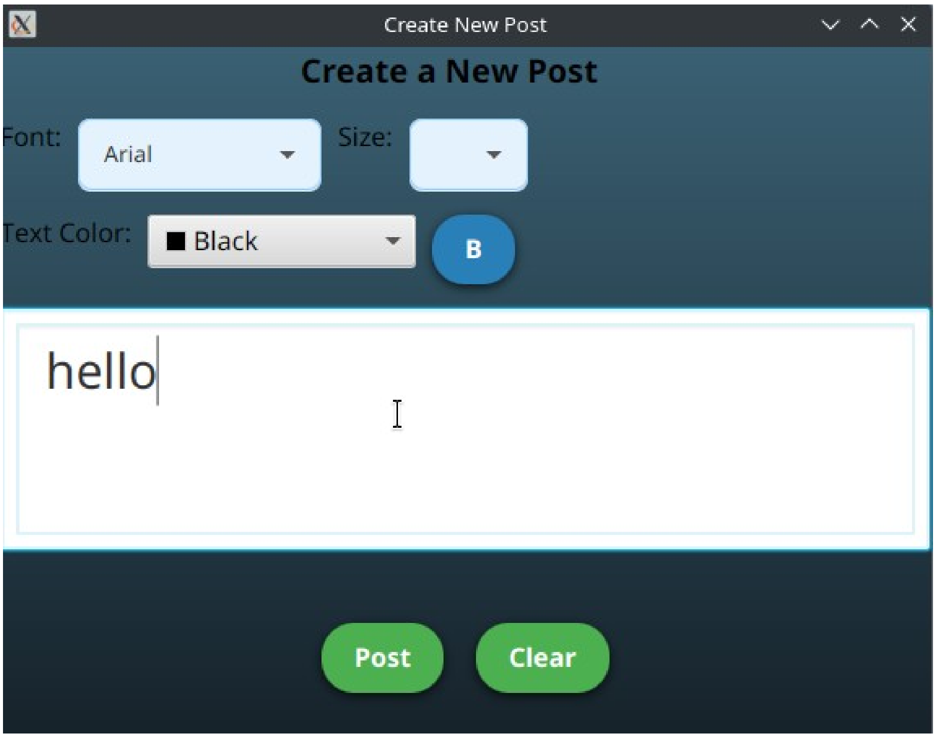


Figure 9.12: MyFriends scene

13.Create Post screen: This scene represents the create post page where user can prepare and send a post to feed.



Font: Arial

Size:

Text Color: ■ Black

hello

Figure 9.13: Create Post scene

14.Visualize Connections screen: This scene represents the visualize connections page , user can visualize his friends. .

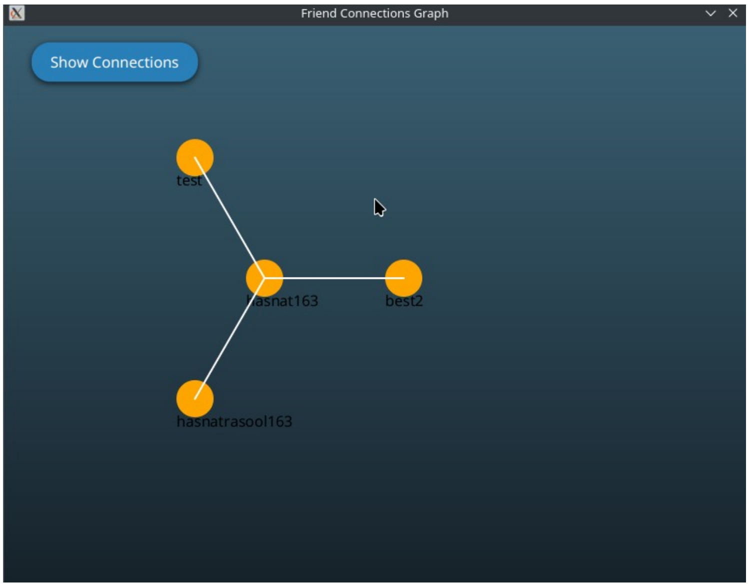


Figure 9.14: Visualize Connections scene

15.Admin Dashboard screen: This scene represents the admin dashboard where admin can navigate to manage users , connecions , posts and visualize network of different users.

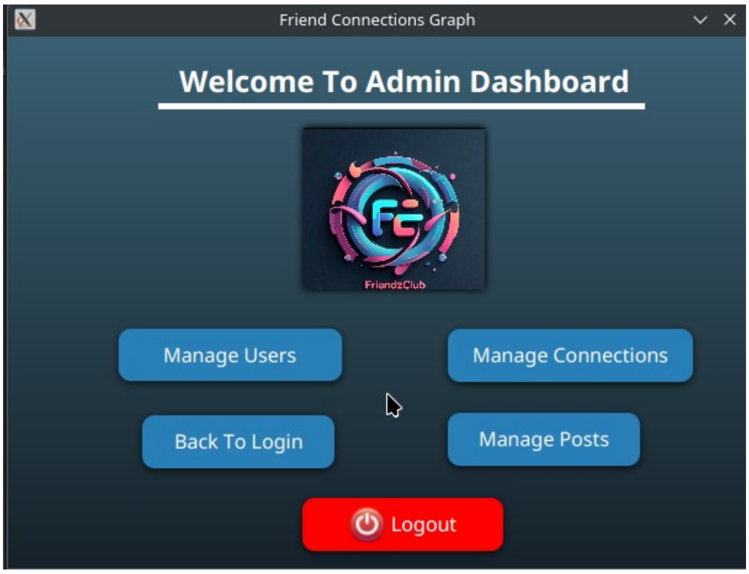
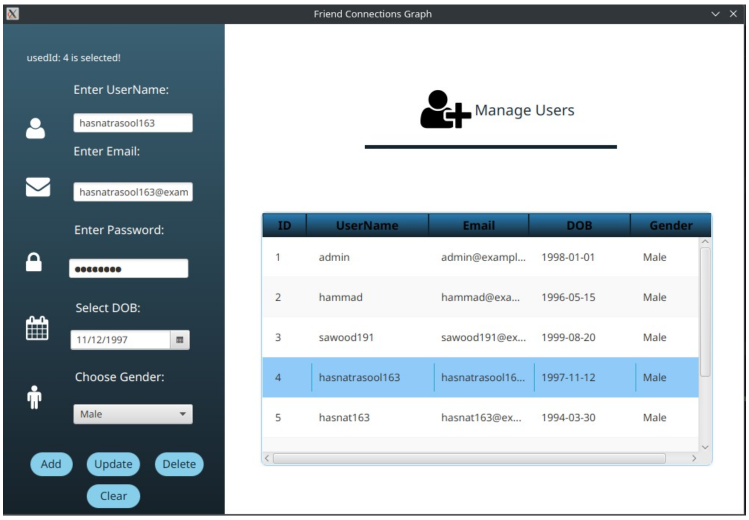


Figure 9.15: Admin Dashboard scene

16.Manage Users screen: This scene represents the manage users page where admin can add user , update user info , or delete a user .



usedId: 4 is selected!

Enter UserName: hasnatrasool163

Enter Email: hasnatrasool163@exam

Enter Password: ●●●●●●●●●

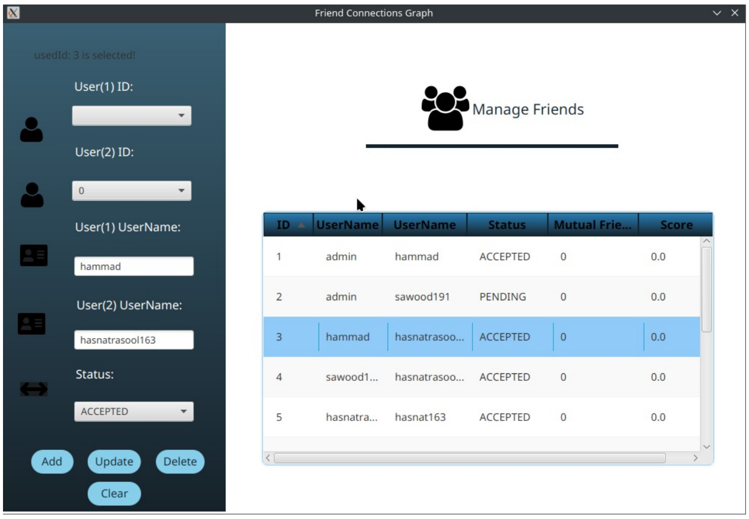
Select DOB: 11/12/1997

Choose Gender: Male

ID	UserName	Email	DOB	Gender
1	admin	admin@exampl...	1998-01-01	Male
2	hammad	hammad@exa...	1996-05-15	Male
3	sawood191	sawood191@ex...	1999-08-20	Male
4	hasnatrasool163	hasnatrasool16...	1997-11-12	Male
5	hasnat163	hasnat163@ex...	1994-03-30	Male

Figure 9.16: Manage Users scene

17.Manage Friends screen: This scene represents the manage friends page where admin can manage friendship status or add new friend request , update or delete .



usedId: 3 is selected!

User(1) ID:

User(2) ID: 0

User(1) UserName: hammad

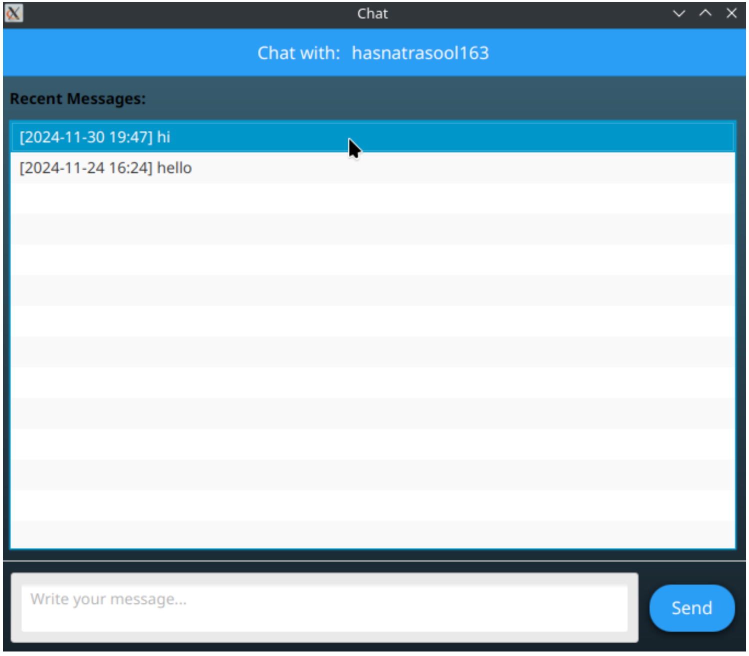
User(2) UserName: hasnatrasool163

Status: ACCEPTED

ID	UserName	UserName	Status	Mutual Frie...	Score
1	admin	hammad	ACCEPTED	0	0.0
2	admin	sawood191	PENDING	0	0.0
3	hammad	hasnatrasoo...	ACCEPTED	0	0.0
4	sawood1...	hasnatrasoo...	ACCEPTED	0	0.0
5	hasnatra...	hasnat163	ACCEPTED	0	0.0

Figure 9.17: Manage Friends scene

18.Chat screen: This scene represents the direct 1-1 chat page where user can send message to his selected friend and view last recent messages.



Chat with: hasnatrasool163

Recent Messages:

[2024-11-30 19:47] hi

[2024-11-24 16:24] hello

Write your message...

Figure 9.18: Chat scene

CONCLUSION

The project entitled as **Social Network Friend Recommendation System** is the system that use graph theory concepts for friend-recommendation.

This project is successfully implemented with all the features mentioned in system requirements specification.

The application provides appropriate information to users according to the chosen service.

The project is designed keeping in view real world scenario and requirements.

REFERENCES

Books and Websites:

- Java Software Structures - Designing and Using Data Structures, 4th edition, Lewis and Chase.
- Code for Interview YouTube Channel.
- BroCode YouTube Channel.
- Database System Concepts, by Silberschatz, Sudarshan, and Korth.
- Fundamentals of Database Systems, RamezElmasri and Shamkant B. Navathe, 7th Edition. 2017, Pearson...