VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", MACHHE, BELAGAVI – 590018

Mini Project Report
on

SOCIAL NETWORKING CLONE

Submitted in partial fulfillment of the requirements for the V semester
Bachelor of Engineering
in
Computer Science and Engineering
of
Visvesvaraya Technological University, Belagavi
by

Mr. Harish Prasad (1CD18CS045)
Mr. Ishan Dubey (1CD18CS048)
Mr. Sumit Anand (1CD18CS177)

Under the Guidance of

Ms Sonia Devi
Assistant Professor,
Dept of CSE,CITech

Ms Priyadarshini M
Assistant Professor,
Dept of CSE,CITech

Department of Computer Science and Engineering
CAMBRIDGE INSTITUTE OF TECHNOLOGY, BENGALURU – 560036
2020-2021

CAMBRIDGE INSTITUTE OF TECHNOLOGY

K.R. Puram, Bengaluru – 560036

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

CERTIFICATE

Certified that **Mr. Harish Prasad, Mr. Ishan Dubey and Mr. Sumit Anand** bearing USN **1CD18CS045, 1CD186CS048 and 1CD18CS177** respectively are bonafide students of **Cambridge Institute of Technology,** has successfully completed Mini Project entitled "**Social NETWORKING Clone**" in partial fulfillment of the requirements for V semester **Bachelor of Engineering** in **Computer Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year 2020 - 2021. It is certified that all Corrections / Suggestions indicated for Internal Assessment have been incorporated in the report. The Database Project report has been approved as it satisfies the academic requirements in respect of Project Work prescribed for the said semester.

Internal Guides:

1) Ms. Sonia Devi
Dept. of CSE., CiTech.

2) Ms. Priyadarshini M.
Dept. of CSE.,
CiTech.

Head of the Dept.

Dr. Shashikumar D. R.
Dept. of CSE., CiTech.

Examiners:

1)

2)

ACKNOWLEDGEMENT

We are extremely thankful to **Dr. Suresh L.,** Principal, CITech., Bengaluru, for providing us the academic ambience and everlasting motivation to carry out this work and shaping our careers.

We express our sincere gratitude to **Dr. Shashikumar D. R.,** HOD, Dept. of Computer Science and Engineering, CITech., Bengaluru, for his stimulating guidance, continuous encouragement and motivation throughout the course of present work.

We also wish to extend our thanks to **Ms. Sonia Devi,** Assistant Professor Dept. of Computer Science and Engineering, CITech., Bengaluru and **Ms. Priyadarshini M.,** Assistant Professor, Dept. of Computer Science and Engineering, CITech., Bengaluru, for their expert guidance and constructive suggestions to improve the quality of this work.

We would also like to thank all other teaching and technical staffs of Department of Computer Science and Engineering, who have directly or indirectly helped us in the completion of this Project Work.

And lastly, we would hereby acknowledge and thank our parents who have been a source of inspiration and also instrumental in the successful completion of this project.

Harish Prasad(1CD18CS045)

Ishan Dubey(1CD18CS048)

Sumit Anand(1CD18CS177)

ABSTRACT

Database management system is a system is a computer software program that is designed as the means of managing all databases that are currently installed on a system hard drive or network. Databases have been in use since the earliest days of electronic computing. A Database Management System (DBMS) is a set of computer programs that controls the creation, maintenance, and the use of a database. It allows organizations to place control of database development in the hands of database administrators (DBAs) and other specialists. A DBMS is a system software package that helps the use of integrated collection of data records and files known as databases. It allows different user application programs to easily access the same database. DBMSs may use any of a variety of database models, such as the network model or relational model. It provides facilities for controlling data access, enforcing data integrity, managing concurrency, and restoring the database from backups. A DBMS also provides the ability to logically present database information to users.

CONTENTS

CHAPTER 1

INTRODUCTION

A social networking service (also social networking site or social media) is an online platform which people use to build social networks or social relationships with other people who share similar personal or career interests, activities, backgrounds or real-life connections. Social networking has become a significant base for marketers seeking to engage customers.

Social networking services vary in format and the number of features. They can incorporate a range of new information and communication tools, operating on desktops and on laptops, on mobile devices such as tablet computers and smartphones.

At the barebones level a social networking site is a website (on desktops) or an application (on mobile phones) meant to store the details useful enough to let the user connect to another user.

Implementation of a contemporary social networking websites is what this project intends to highlight and proposes a user to attain an understanding of behind the stage operation of the same.

1.1 Problem Definition

This project on Social Networking brings in the ease to the people to meet and connect to the people of different demography, race, views and position in a society.
Doing the same using conventional techniques of maintaining a phone book, use of letters to communicate to people (known) or else it was quite uneasy to people to start interacting with unknown people they might find worth meeting. The problem being small on paper is enormous. The problem of maintaining the data and the corresponding identity, storage of this data, feasibility of procuring information form it and most crucial security of the data holding system.

1.2 Need

We are glad to inform…

That the solutions to the above given problems were detected and solved using the Database Management Solutions which we have brought here in this project.

Connecting to people was difficult, this is where the social networking sites and services fill the gap. With the facility of internet this was made possible. But this ease brought along with it, data of each individual and his contacts.

Enters Database Management System. Connecting the web application with a storage system (here Database) proved to be safest as well as efficient option compared to traditional file handling system

We required a system such as which looked pleasing on the front-end but was robust behind the curtains.

The kind of bullet points that were concerning in the view of this targeted system was with the use of a combination of webpages and a database.

The webpages will be the objects interacting with the user. The webpage on the other hand would stream the data to the DBMS manager. It is the then the duty of the DBMS manager to feed the received data to the database, as well as procure the data as and when requested by the webpage (i.e. The user).

CHAPTER 2

REQUIREMENTS

2.1 Software Requirement Specifications

Operating System : Ubuntu 20.04 LTS (Focal Fossa)

Front End : HTML5, CSS & JSP

Back End Server : Tomcat 9.0

Database Management System : Oracle 11g V2

Documentation : LibreOffice 7.0

2.2 Hardware Requirement Specifications

Computer Processor : Core i3 Processor Speed 2.3 GHz

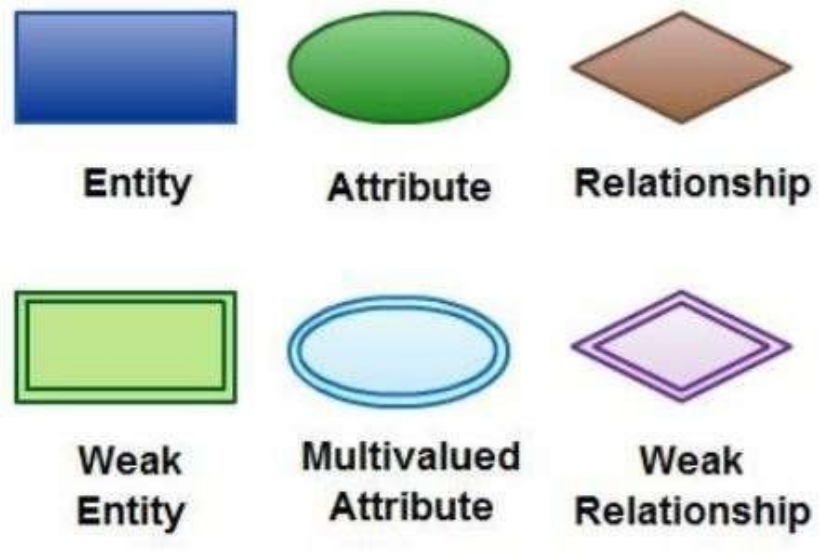
RAM : 4Gb

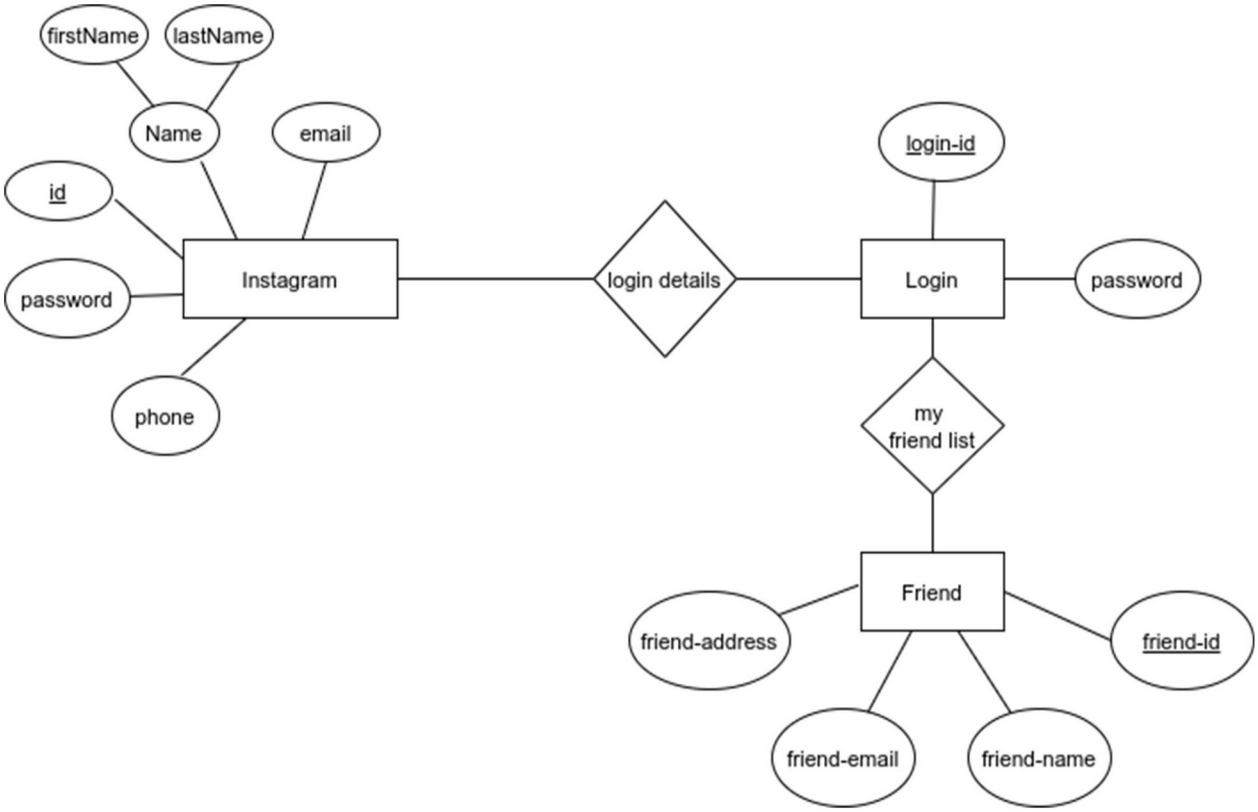
Hard Disk : 100Gb

CHAPTER 3

ENTITY RELATIONSHIP DIAGRAM

An Entity Relationship Diagram (ERD) is a visual representation of different entities within a system and how they relate to each other. Although data modeling has become a necessity around 1970's there was no standard way to model databases or business processes. Although many solutions were proposed and discussed none were widely adopted. Peter Chen is credited with introducing the widely adopted ER model in his paper "The Entity Relationship Model Toward a Unified View of Data". The focus was on entities and relationships and he introduced a diagramming representation for database design as well. They are widely used to design relational databases. The entities in the ER schema become tables, attributes and converted the database schema. Since they can be used to visualize database tables and their relationships it's commonly used for database troubleshooting as well. Entity relationship diagrams are used in software engineering during the planning stages of the software project. They help to identify different system elements and their relationships with each other. It is often used as the basis for data flow diagrams or DFD's as they are commonly known.




firstName
lastName
Name
email
id
password
phone
Instagram
login details
login-id
Login
password
my
friend list
Friend
friend-address
friend-id
friend-email
friend-name

CHAPTER 4

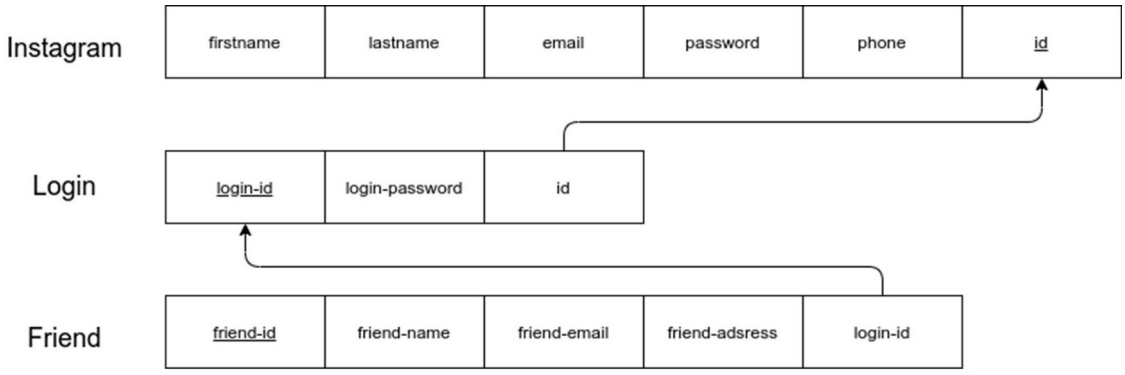
SCHEMA DIAGRAM

4.1 SCHEMA DIAGRAM

A database schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data. A database schema defines its entities and the relationship among them. It contains a descriptive detail of the database, which can be depicted by means of schema diagrams. It's the database designers who design the schema to help programmers understand the database and make it useful. It is important that we distinguish these two terms individually. Database schema is the skeleton of database. It is designed when the database doesn't exist at all. Once the database is operational, it is very difficult to make any changes to it. A database schema does not contain any data or information. A database instance is a state of operational database with data at any given time. It contains a snapshot of the database. Database instances tend to change with time. A DBMS ensures that its every instance (state) is in a valid state, by diligently following all the validations, constraints, and conditions that the database designers have imposed. A database schema can be divided broadly into two categories –

• Physical Database Schema − This schema pertains to the actual storage of data and its form of storage like files, indices, etc. It defines how the data will be stored in a secondary storage.

• Logical Database Schema − This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints.


Instagram
firstname
lastname
email
password
phone
id
Login
login-id
login-password
id
Friend
friend-id
friend-name
friend-email
friend-adsress
login-id

CHAPTER 5

IMPLEMENTATION

5.1 Backend Implementation

```
create table instagram
(
firstname varchar2(10),
lastname varchar2(10),
email varchar2(40),
password varchar2(20),
phone varchar2(15),
id number(3) primary key
);

create table login
(
login_id varchar(30) primary key,
login_password varchar2(20),
id number(10),
foreign key(id) references instagram(id)
);

create table friend
(
friend_id number(5) primary key,
friend_name varchar2(20),
friend_name varchar2(50),
friend_name varchar2(30),
login_id varchar2(30),
foreign key(login_id) references login(login_id)
);
```

Connectivity :

```
package com.hpk.instagram.dao;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

import java.sql.ResultSet;


public class InstagramDAO implements InstagramDAOInterface {


        public int createProfile(String firstName, String secondName, String email, String
password, String phone) {
```

```
int i = 0;
int i1 = 0;

try {
        Class.forName("oracle.jdbc.driver.OracleDriver");
        Connection con = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","harish","hpk");
        PreparedStatement ps = con.prepareStatement("insert into instagram (firstName,lastName,email,password,phone) values(?,?,?,?,?)");
        ps.setString(1,firstName);
        ps.setString(2,secondName);
        ps.setString(3,email);
        ps.setString(4,password);
        ps.setString(5,phone);

        i = ps.executeUpdate();
        con.close();
        Connection con1 = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","harish","hpk");
        PreparedStatement ps2 = con1.prepareStatement("select * from Instagram");

        ResultSet res = ps2.executeQuery();
        int c = 0;
        while(res.next()) {
                c = res.getInt("id");
        }
        con.close();
        Connection con2 = DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","harish","hpk");

        if(i > 0) {
                PreparedStatement ps1 = con2.prepareStatement("insert into login values (?,?,?)" );

                ps1.setString(1,email);
```

```
                    ps1.setString(2,password);
                    ps1.setInt(3,c);
                    i1= ps1.executeUpdate();
                }

            }
            catch(Exception e) {
                e.printStackTrace();
            }

            return i1;
        }

        public boolean loginProfile(String email,String password) {

            boolean b = false;
            try {
                Class.forName("oracle.jdbc.driver.OracleDriver");
                Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","harish","hpk");
                PreparedStatement ps = con.prepareStatement("select * from login where
login_id = ? and login_password = ?");
                ps.setString(1, email);
                ps.setString(2, password);
                ResultSet res = ps.executeQuery();
                if(res.next()) {
                    b = true;
                }

            }

            catch(Exception e) {
                e.printStackTrace();
            }
            return b;
```

```
	}

	public ResultSet viewProfile(String e)
		{ ResultSet res = null;
		System.out.println(e);
		try {
			Class.forName("oracle.jdbc.driver.OracleDriver");
			Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","harish","hpk");

			PreparedStatement ps = con.prepareStatement("select * from instagram
i,login l where i.id=l.id and l.login_id = ? ");
			ps.setString(1, e);

			 res = ps.executeQuery();
//			 System.out.println(res.next());



		}

		catch(Exception er) {
			er.printStackTrace();
		}
		return res;
	}

	public ResultSet viewFriendList(String e) {

		ResultSet res = null;
		try {
			System.out.println(e);
			Class.forName("oracle.jdbc.driver.OracleDriver");
			Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","harish","hpk");
```

```
                PreparedStatement ps = con.prepareStatement("select * from friend f,login
l where l.login_id = f.login_id and l.login_id = ? ");
                ps.setString(1, e);
                res = ps.executeQuery();
//                 System.out.println(res.next());

        }

        catch(Exception er) {
                er.printStackTrace();
        }
        return res;



    }

    public int deleteProfile(String id)
        { int i = 0;
        try {
                Class.forName("oracle.jdbc.driver.OracleDriver");
                Connection con =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","harish","hpk");

                PreparedStatement ps = con.prepareStatement("delete from friend  where
login_id =  ?");
                ps.setString(1, id);
                ps.executeUpdate();

                PreparedStatement ps1 = con.prepareStatement("delete from login where
login_id =  ?");
                ps1.setString(1, id);
                ps1.executeUpdate();

                PreparedStatement ps2 = con.prepareStatement("delete from instagram
where email = ?");
```

```
                ps2.setString(1, id);
                i = ps2.executeUpdate();
            }
            catch(Exception er) {
                er.printStackTrace();
            }
            return i;
        }
        public int editProfile(String firstName, String lastName, String email, String password,
String phoneNumber) {
            int i = 0;
            try {                   //step1:load driver
                Class.forName("oracle.jdbc.driver.OracleDriver");
                //step2:create connection with database
                Connection                              con                             =
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","harish","hpk");
                //step3:create query: this is dynamic type of query
                PreparedStatement ps=con.prepareStatement("update instagram set
firstname=?, lastname=?, password=?,phone=? where email=?");
                ps.setString(1, firstName);
                ps.setString(2, lastName);
                ps.setString(3, password);
                ps.setString(4, phoneNumber);
                ps.setString(5, email);
                //execute query
                i=ps.executeUpdate();          //return type is int ; for
                System.out.println(i);
                                }
            catch(Exception e){
                e.printStackTrace();
            }
            return i;
        }}
```

5.2 Frontend Implementation

```
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link rel="icon" href="../../../../favicon.ico">
    <title>siteMe</title>

    <!-- Bootstrap core CSS -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>

   <!-- Custom styles for this template -->
   <link href="css/carousel.css" rel="stylesheet">
   <link href="css/Wendy.css" rel="stylesheet">
  </head>
<body>

  <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <a class="nav-identity" href="index.html"><img src="https://en.instagram-brand.com/wp-
content/themes/ig-branding/prj-ig-branding/assets/images/ig-logo.svg" class="nav-
identity_logo" alt="Instagram" width="36" height="36"></a><br>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarSupportedContent"
            aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle
navigation">
                  <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      <ul class="navbar-nav mr-auto">
        <li class="nav-item">
        <a class="nav-link active" aria-current="page" href="./index.html">Home</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="./about.html">About</a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="./contact.html">Contact Us</a>
        </li>
      </ul>
     <div style="padding:5" id="signin">
      <button type="button" class="btn btn-outline-success" data-toggle="modal" data-
target="#loginModal">
        Sign In
      </button>

      <button type="button" class="btn btn-outline-primary" data-toggle="modal" data-
target="#registerModal">
```

```
          Sign Up
         </button>

        </div>
       </div>
   </nav>
   <br>

<!-- Register Modal -->

   <div class="modal fade" id="registerModal" aria-hidden="true">
     <div class="modal-dialog" role="document">
       <div class="modal-content">
         <div class="modal-header">
           <h5 class="modal-title" id="exampleModalLabel">Join Us & get Connected!</h5>
           <button type="button" class="close" data-dismiss="modal" aria-label="Close">
             <span aria-hidden="true">&times;</span>
           </button>
         </div>
         <div class="modal-body">
          <form class="form-signin" action="InstagramController" method="post"
name="regform">
             <input type="text" class="form-control" placeholder="First name" name="fname"
required autofocus />
             <br>
             <input type="text" class="form-control" placeholder="Surname" name="sname" />
             <br>
             <input type="email" class="form-control" placeholder="Email address" name="email"
onblur="startAjax();"
               required />
             <br>
             <input type="password" class="form-control" placeholder="Password"
name="password" required>
             <br>
             <input type="number" id="phn" class="form-control" placeholder="Phone number"
name="phone" maxlength="10"
               required>
         </div>
          <div class="modal-footer">
           <div id="message" style="float:left;position: relative;left: -173px; color:red;"></div>
           <button type="submit" class="btn btn-primary" id="register">Done</button>
           </form>
         </div>
       </div>
     </div>
   </div>

<!-- Login Modal -->

    <div class="modal fade" id="loginModal">
      <div class="modal-dialog" role="document">
```

```
        <div class="modal-content">
          <div class="modal-header">
            <h5 class="modal-title" id="exampleModalLabel">Please Sign in! </h5>
            <button type="button" class="close" data-dismiss="modal" aria-label="Close">
              <span aria-hidden="true">&times;</span>
            </button>
          </div>
          <div class="modal-body">
            <div class="container">
              <form class="form-signin" action="InstagramLogin" method="post"
name="loginform">
                <input type="email" id="inputEmail" class="form-control" placeholder="Email
address" name="email" required
                  autofocus>
                <br>
                <input type="password" id="inputPassword" class="form-control"
placeholder="Password" name="password"
                  required>
                <br>
            </div>
          </div>
          <div class="modal-footer">
            <div id="message1" style="float:left;color:red;"></div>
            <label><a href="#">Forgot Password?</a></label>
            <button class="btn btn-success" type="submit"">Sign in</button>
            </form>
          </div>
        </div>
      </div>
    </div>

<!-- Slider -->

    <div id="myCarousel" class="carousel slide" data-ride="carousel">
      <ol class="carousel-indicators">
        <li data-target="#myCarousel" data-slide-to="0" class="active"></li>
        <li  data-target="#myCarousel"  data-slide-to="1"></li>
        <li  data-target="#myCarousel"  data-slide-to="2"></li>
      </ol>
      <div class="carousel-inner">
        <div class="carousel-item active">
          <img class="first-slide" src="images/1.jpg" alt="First slide">
          <div class="container">
            <div class="carousel-caption">
              <br>
            </div>
          </div>
        </div>
        <div class="carousel-item">
          <img class="second-slide" src="images/2.jpg" alt="Second slide">
          <div class="container">
```

```
          <div class="carousel-caption text-right">
          </div>
        </div>
      </div>
      <div class="carousel-item">
        <img class="third-slide" src="images/3.jpg" alt="Third slide">
        <div class="container">
          <div class="carousel-caption text-right">
          </div>
        </div>
      </div>
    </div>
    <a class="carousel-control-prev" href="#myCarousel" role="button" data-slide="prev">
      <span class="carousel-control-prev-icon" aria-hidden="true"></span>
      <span class="sr-only">Previous</span>
    </a>
    <a class="carousel-control-next" href="#myCarousel" role="button" data-slide="next">
      <span class="carousel-control-next-icon" aria-hidden="true"></span>
      <span class="sr-only">Next</span>
    </a>
    <br>
    <br>
  </div>


   <!-- Marketing messaging and featurettes
     ================================================== -->
   <!-- Wrap the rest of the page in another container to center all the content. -->

     <div class="container marketing">
    <!-- Three columns of text below the carousel -->
    <div class="row">
      <div class="col-lg-4">
        <div class="card" style="width: 18rem;">
                                <img class="card-img-top" src="./images/4.jpg" alt="Card image
cap">
                                 <div class="card-body">
                                <h5 class="card-title"><strong>Feed</strong></h5>
                                <p class="card-text">Some quick example text to build on the card
title and make up the bulk of the card's content.</p>
                                <a href="#" class="btn btn-primary">Read More</a>
                                </div>
                        </div>
      </div><!-- /.col-lg-4 -->
      <div class="col-lg-4">
        <div class="card" style="width: 18rem;">
                                <img class="card-img-top" src="./images/6.jpg" alt="Card image
cap">
                                 <div class="card-body">
                                <h5 class="card-title"><strong>Private Chat</strong></h5>
```

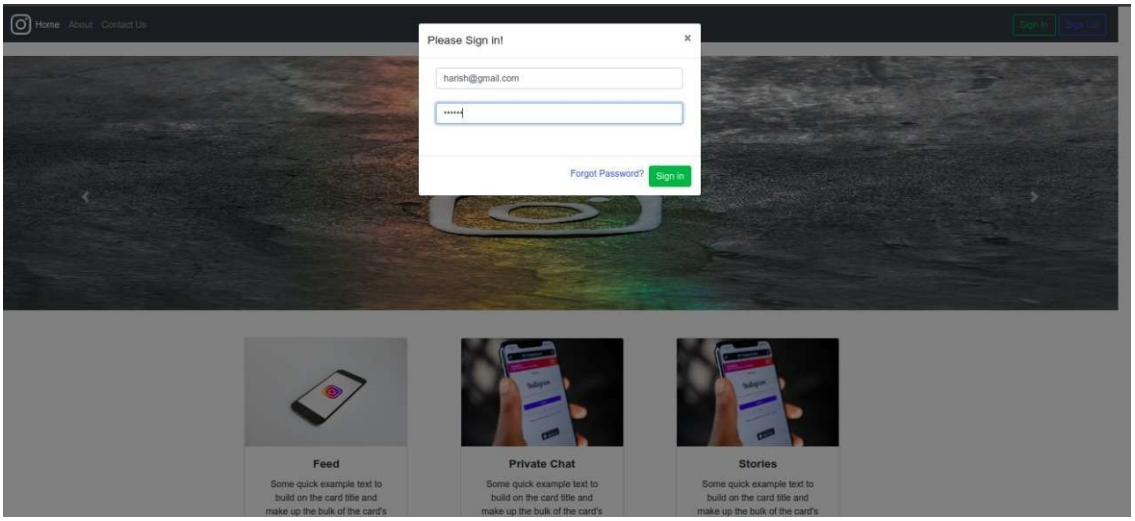
```
                                <p class="card-text">Some quick example text to build on the card
title and make up the bulk of the card's content.</p>
                                <a href="#" class="btn btn-primary">Read More</a>
                                </div>
                        </div>
                </div>


        <!-- /.col-lg-4 -->
        <div class="col-lg-4">
          <div class="card" style="width: 18rem;">
                                <img class="card-img-top" src="./images/6.jpg" alt="Card image
cap">
                                 <div class="card-body">
                                <h5 class="card-title"><strong>Stories</strong></h5>
                                <p class="card-text">Some quick example text to build on the card
title and make up the bulk of the card's content.</p>
                                <a href="#" class="btn btn-primary">Read More</a>
                                </div>
                        </div>
        </div>
       </div>
       </div>
       <br>
       <br>
       <br>

<div class="container marketing">
       <!-- Three columns of text below the carousel -->
       <div class="row">
        <div class="col-lg-4">
          <div class="card" style="width: 18rem;">
                                <img class="card-img-top" src="./images/4.jpg" alt="Card image
cap">
                                 <div class="card-body">
                                <h5 class="card-title"><strong>Make Reals</strong></h5>
                                <p class="card-text">Some quick example text to build on the card
title and make up the bulk of the card's content.</p>
                                <a href="#" class="btn btn-primary">Read More</a>
                                </div>
                        </div>
       </div><!-- /.col-lg-4 -->
       <div class="col-lg-4">
          <div class="card" style="width: 18rem;">
                                <img class="card-img-top" src="./images/6.jpg" alt="Card image
cap">
                                 <div class="card-body">
                                <h5 class="card-title"><strong>Make Friends</strong></h5>
                                <p class="card-text">Some quick example text to build on the card
title and make up the bulk of the card's content.</p>
                                <a href="#" class="btn btn-primary">Read More</a>
```

```
                                </div>
                            </div>
                        </div>


        <!-- /.col-lg-4 -->
        <div class="col-lg-4">
          <div class="card" style="width: 18rem;">
                                <img class="card-img-top" src="./images/6.jpg" alt="Card image
cap">
                                 <div class="card-body">
                                <h5 class="card-title"><strong>Play Games</strong></h5>
                                <p class="card-text">Some quick example text to build on the card
title and make up the bulk of the card's content.</p>
                                <a href="#" class="btn btn-primary">Read More</a>
                                </div>
                            </div>
        </div>
      </div>
      </div>
      <br>
      <br>
      <br>

      <!-- FOOTER -->
      <footer class="container">
        <p class="float-right"><a href="#">Go Top</a></p>
        <p>&copy; 2020 <strong>SiteMe</strong> &middot; <a href="#">Privacy</a> &middot;
<a href="#">Terms</a></p>
      </footer>

      <!-- Bootstrap core JavaScript
      ================================================== -->
      <!-- Placed at the end of the document so the pages load faster -->
      <script src="js/jquery-3.2.1.slim.min.js"
        integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
        crossorigin="anonymous"></script>
      <script>window.jQuery || document.write('<script src="../../../../assets/js/vendor/jquery-
slim.min.js"><\/script>')</script>
      <script src="../../../../assets/js/vendor/popper.min.js"></script>
      <script src="js/bootstrap.min.js"></script>
      <!-- Just to make our placeholder images work. Don't actually copy the next line! -->
      <script src="js/holder.min.js"></script>

   </body>
</html>
```

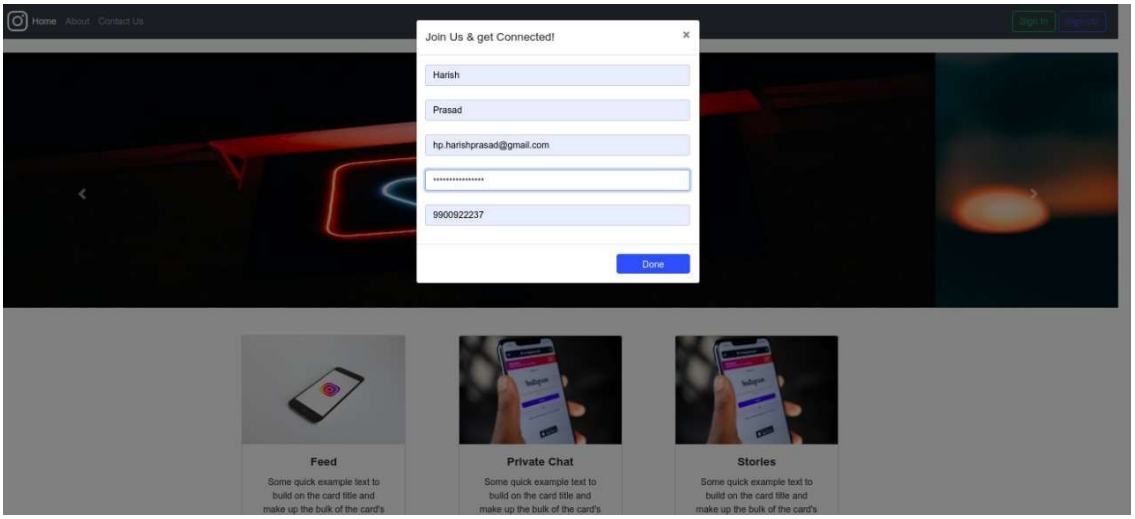
CHAPTER 6

SNAPSHOTS

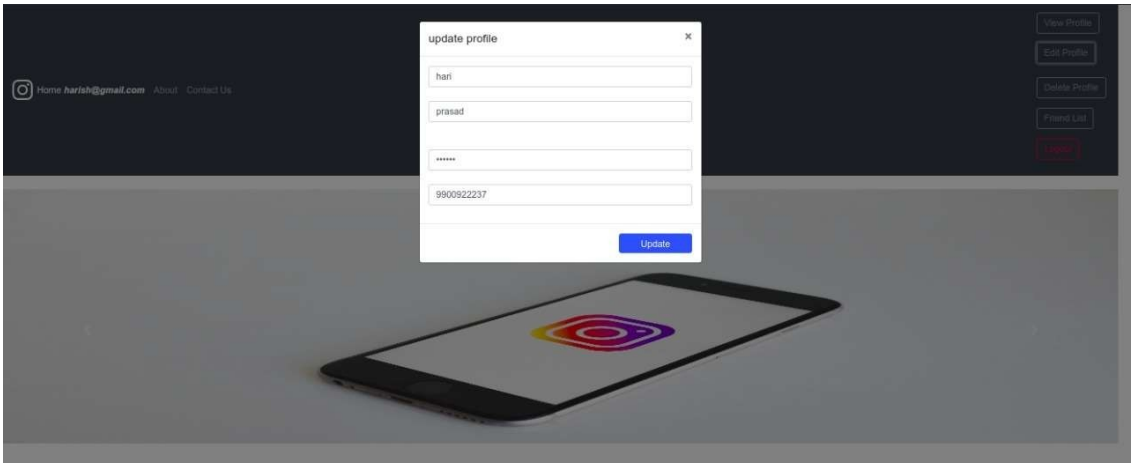
6.1 Login Page



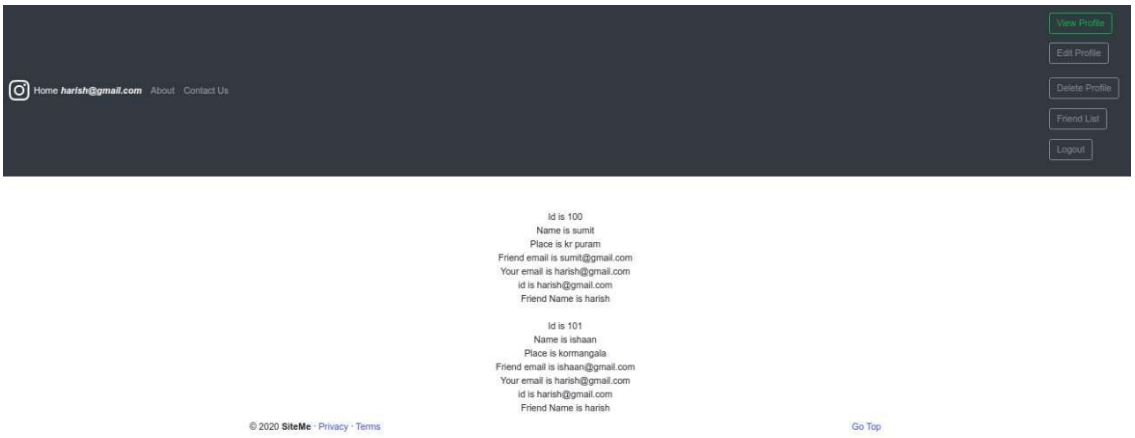
6.2 Insertion



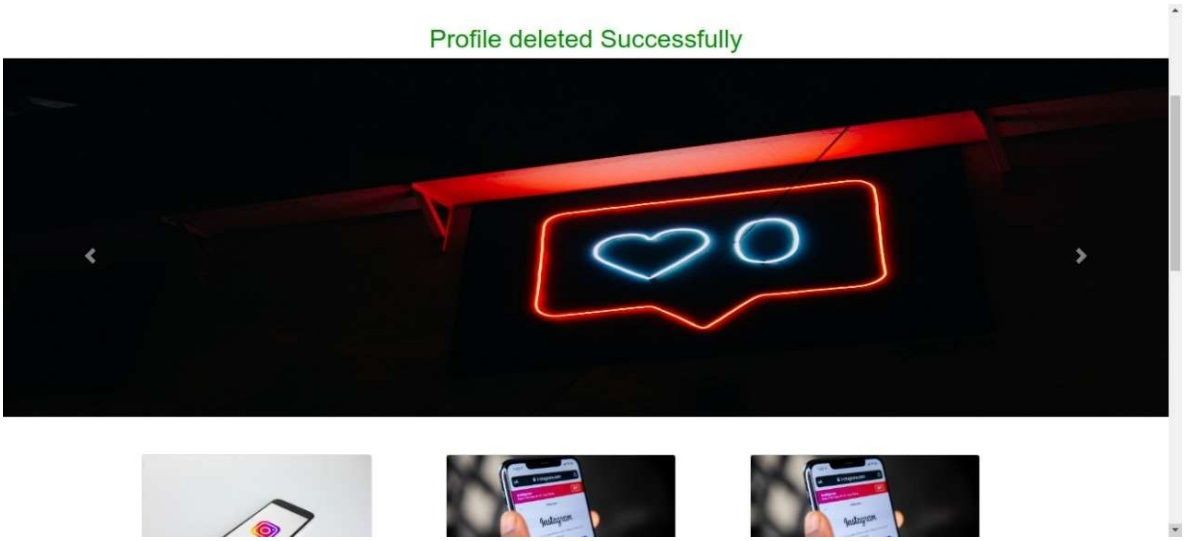
6.3 Updation



6.4 Viewing



6.5 Deletion



6.6 Records

```
harish@hpk:~
Welcome Back Harish: rlwrap sqlplus

SQL*Plus: Release 11.2.0.2.0 Production on Mon Jan 18 00:19:44 2021

Copyright (c) 1982, 2011, Oracle.  All rights reserved.

Enter user-name: harish
Enter password:

Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL> set linesize 300;
SQL> select * from instagram;

FIRSTNAME  LASTNAME   EMAIL                                    PASSWORD             PHONE    ID
---------- ---------- ---------------------------------------- -------------------- ---------- ----------
hari       prasad     harish@gmail.com                         harish               9900922237            436
sumit      anand      sumit@gmail.com                          sumit                7676737231            443
ishaan     dubey      ishaan@gmail.com                         ishaan               8050755077            450
mohan      kumar      mohan@gmail.com                          mohan                9241135113            457

SQL> select * from login;

LOGIN_ID                       LOGIN_PASSWORD                 ID
------------------------------ ------------------------------ ----------
harish@gmail.com               harish                        436
sumit@gmail.com                sumit                         443
mohan@gmail.com                mohan                         457
ishaan@gmail.com               ishaan                        450

SQL> select * from friend;

 FRIEND_ID FRIEND_NAME          FRIEND_ADDRESS                                     FRIEND_EMAIL                   LOGIN_ID
---------- -------------------- -------------------------------------------------- ------------------------------ ------------------------------
-------------
       100 sumit                kr puram                                           sumit@gmail.com                harish@gmail.com
       101 ishaan               kormangala                                         ishaan@gmail.com               harish@gmail.com

SQL>
```

CONCLUSION

This project enables the user to attain a basic understanding of the under the bonnet working of a social networking website. It promotes the user to gain curiosity as to the extent of how the simple working of a site is scaled to such an extent to provide services to mass magnitude of network of tangible (computers) as well as intangible (emotional and economic) components of the living ecosystem.

The project has even encouraged us, being the creators of this project to research further into this area of networking. Its inner and upper-level implementation, how it is scaled, how the infrastructure is maintained.

This project has guided our path through various aspects of computer science in networking and communications where developing online application plays a major role.

REFERENCES

[1] Herbert Schildt, Java The Complete Reference, 7th Edition, Tata McGraw Hill, 2007.

[2] Fundamentals of Database Systems, Ramez Elmasri and Shamkant B. Navathe, 7th Edition, 2017, Pearson

[3] HTML and CSS: Design and build websites, by Jon Duckett

[4] Learning Web Design: A beginner's guide to HTML, CSS, Javascript, and Web Graphics, By Jennifer Niederst Robbins

[5] JavaScript: The Good Parts : Douglas Crockford

CAMBRIDGE INSTITUTE OF TECHNOLOGY

K. R. PURAM, BENGALURU - 560036

Vision

To become a premier institute transforming our students to be global professionals.

Mission

M1: Develop competent Human Resources, and create state-of-the-art infrastructure to impart quality education and to support research.

M2: Adopt tertiary approach in teaching – learning pedagogy that transforms students to become professionally competent technocrats and entrepreneurs.

M3: Nurture and train students to develop the qualities of global professionals.

Department of Computer Science and Engineering

Vision

To impart quality education in the field of Computer Science and Engineering with emphasis on innovative thinking, communication and leadership skills to meet the global challenges in IT paradigm.

Mission

M1: Focus on student centric approach through experiential learning and necessary infrastructure.

M2: Develop innovative thinking, communication and leadership skills by creating conducive environment and relevant training.

M3: Enrich students by developing the traits of global professionals.