# CHAPTER 1

## INTRODUCTION

The COLLEGE SOCIAL NETWORK is a great project. This project is highly interactive and easy to use and understand. This is an online platform which can be used by the college students to build social relationship with other students and faculties who share similar personal or career interests, activities, backgrounds or real-life connections. This website can be accessed using a web browser. This platform allows users to communicate with others, share their individual ideas, activities and events. Users can register themselves by entering their names, unique usernames and passwords. The passwords are stored in an encrypted format for enhanced security. The registered users can follow each other as per their interests. They can also set their profile pictures and update their bio from time to time. The main aim of this project is to bring the students of the college all at one place and establish a friendly environment inside the college.

* 1. **Need of a College Social Network**

A college can have thousands of students of different streams coming from different places of the country with different cultures, traditions, ideas and thoughts. The students need to interact with each other, but there might be a few students who are not comfortable enough to directly meet someone and talk to them face to face.

A College Social Network can help them in the following ways:

* Students can make more friends within short span of time.
* Students will remain updated with the new and upcoming thoughts and ideas.
* Students will also remain updated about all the academic and non-academic activities taking place inside or outside the college.
* Students can share as well as obtain knowledge by using this website.

# CHAPTER 2

## REQUIREMENT SPECIFICATION

A high-level requirements specification is required. The purpose of the requirements analysis is to identify requirements for the proposed system. The emphasis is on the discovery of user requirements.

### 2.1 SOFTWARE REQUIREMENTS

Operating System : Windows 10 or any compatible operating system.

Database : MySQL

Tools : WAMP Server

### 2.2 HARDWARE REQUIREMENTS

Processor : Any Processor above 500 MHz

RAM : 4GB

Hard Disk : 2 GB free space

Input device : Keyboard, Mouse

Output device : Monitor

System type : 32-bit or 64-bit operating system

### 2.3 FUNCTIONAL REQUIREMENTS

**Home page:** Home page is the first page of the website. Home page contains the Login/Registration page. The registered users can login to the site by entering their unique Usernames and Passwords.

### 2.4 NON-FUNCTIONAL REQUIREMENTS

### PERFORMANCE:

Performance requirements define acceptable response times for system functionality.

* The load time for user interface screens shall take no longer than five seconds.
* The log in information shall be verified within five seconds.
* Queries shall return results within five seconds.

### RELIABILITY:

* Good validations for user inputs will be done
* Avoid incorrect storage of records.

### SECURITY:

* Encrypted Password (makes use of SHA-256 encryption method).

### FLEXIBILITY:

• The system keeps on updating the data according to the user registrations that take place.

### MAINTAINABILITY:

• During maintenance stage, the SRS can be referred for the validation.

### TIMELINESS:

• The system carries out all the operations with consumptions of very less time.

# CHAPTER 3

## OBJECTIVE OF THE PROJECT

The main objective of this project is to create an online platform which can be used by the college students to build social relationship with other students and faculties who share similar personal or career interests, activities, backgrounds or real-life connections.

This project also aims to provide complete user authenticity and data protection. It does not allow two or more users to have the same usernames. The passwords of the users are stored in an encrypted format for a better security. Users can post messages publicly as well as privately. The private messages sent to the intended user can be seen by him only and no one else. The database maintains a well-indexed record of all the registered users, their usernames, passwords, messages, posts and friend counts.

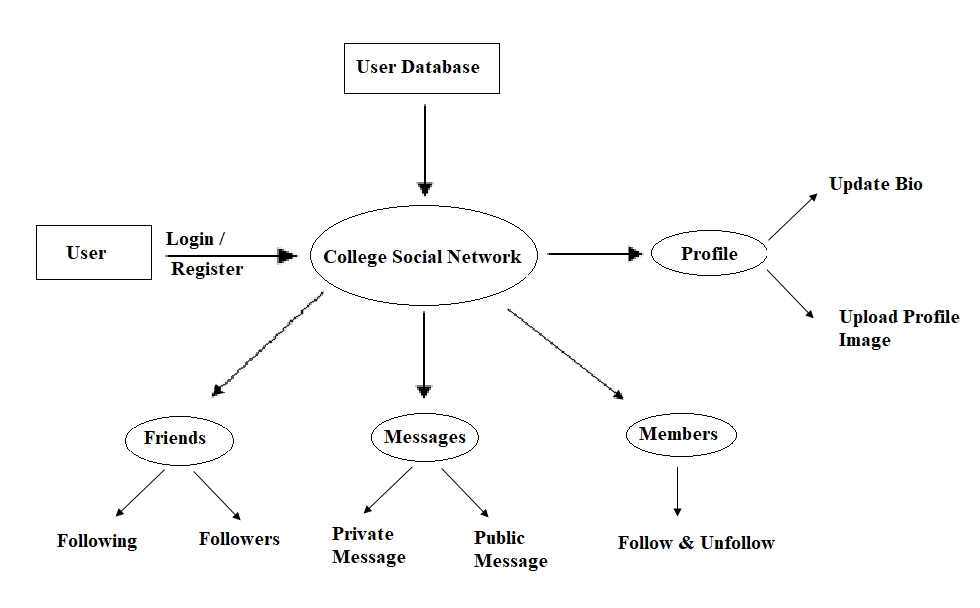
We have made this project user friendly and highly interactive. It tries to bring the students of the college all at one place and establish a friendly environment inside the college.

# CHAPTER 4

## SYSTEM DESIGN

### 4.1 FLOW OF WEB PAGES

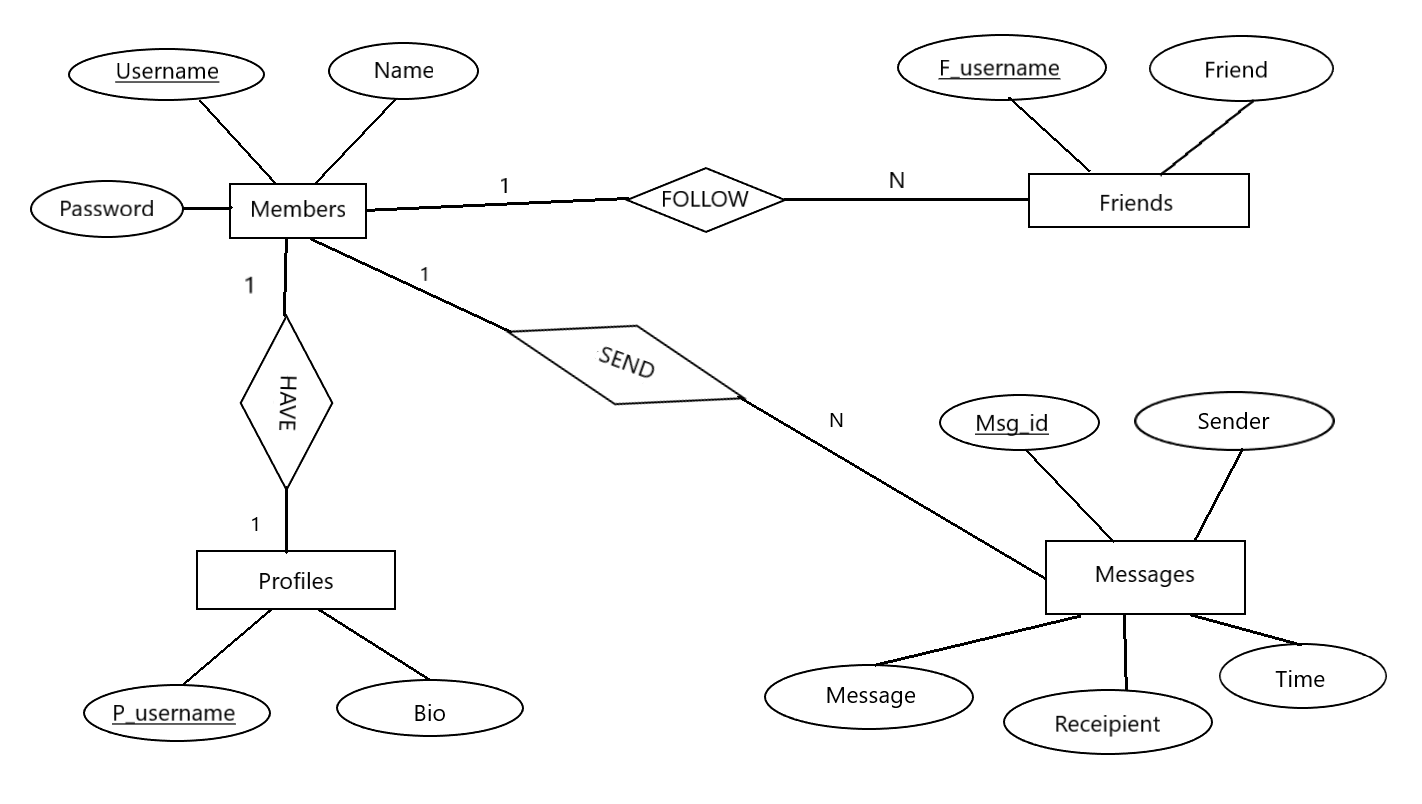
A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different [use cases](https://en.wikipedia.org/wiki/Use_case) in which the user is involved. Figure 4.1 below shows the use case diagram for this website.



**Figure 4.1: Flow of Web Pages**

### 4.2 ENTITY RELATIONSHIP DIAGRAM

The entity-relationship data model is based on a perception of a real world that consists of a collection of basic objects called entities and of relationships among these objects. An entity is an “object” in the real world that is distinguishable from other objects. For e.g. each customer is an entity and rooms can be considered to be entities. Entities are described by a set of attributes. Figure 4.2 Shows the Entity Relationship between the tables.



**Figure 4.2: Entity Relationship Diagram**

# CHAPTER 5

## IMPLEMENTATION

### 5.1 SOURCE CODE

**Index.php**

<!DOCTYPE html>

<html lang="en">

<head>

    <title>Social Network</title>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

<!DOCTYPE html>

<html lang="en">

<head>

    <title>Social Network</title>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" type="text/css" href="css/util.css">

    <link rel="stylesheet" type="text/css" href="css/main.css">

</head>

<body>

<?php

$error = $user = $pass = "";

if (isset($\_POST['user'])) {

$user = sanitizeString($\_POST['user']);

$pass = sanitizeString($\_POST['pass']);

if ($user == "" || $pass == "")

$error = "Not all fields were entered<br>";

else {

$result = queryMySQL("SELECT user,pass FROM members

WHERE user='$user' AND pass='$pass'");

if ($result->num\_rows == 0) {

$error = "<span class='error'>Username/Password

invalid</span><br><br>";

} else {

$\_SESSION['user'] = $user;

$\_SESSION['pass'] = $pass;

die("You are now logged in. Please <a href='members.php?view=$user'>" ."click here</a> to continue.<br><br>");

}

}

}

?>

<form method="post" action="login.php" name="login">

<div class="limiter">

<form class="login100-form validate-form">

Welcome<br>

                        to

                        <H3>Grad Bevy:</H3><br>

                        <h5>College Social Network</h5>

<div class="wrap-input100 validate-input">

<input class="input100" type="text" name="user" placeholder="Username">

</div>

<div class="wrap-input100 validate-input" data-validate="Enter password">

                        <span class="btn-show-pass">

                            <i class="zmdi zmdi-eye"></i>

                        </span>

<input class="input100" type="password" name="pass" placeholder="Password">

                    </div>

                    <div class="container-login100-form-btn">

                            <button class="login100-form-btn">

                                Login

                            </button>

</div>

                        <span class="txt1">

                            Don’t have an account?

                        </span>

                        <a class="txt2" href="signup.php">

                            Sign Up

                        </a>

                    </div>

    </div>

</form>

</body>

</html>

**Header.php**

<?php

session\_start();

echo "<!DOCTYPE html>\n<html><head>";

require\_once 'functions.php';

$userstr = ' (Guest)';

if (isset($\_SESSION['user'])) {

$user = $\_SESSION['user'];

$loggedin = TRUE;

$userstr = " ($user)";

} else $loggedin = FALSE;

?>

<head>

<title>Gradbevy</title>

    <meta charset="utf-8" />

    <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no" />

    <link rel="stylesheet" href="assets/css/main.css" />

<script src='javascript.js'></script>

<script src='/js/bootstrap.js'></script>

<link rel='stylesheet' href='/css/bootstrap.css' type='text/css'>

</head>

<?php

if ($loggedin) {

?>

<body>

<header id="header">

<div class="inner">

<a href="header.php" class="logo">Grad Bevy</a>

<a href='home.php'>Home</a>

<a href='profile.php'>Profile</a>

<a href='messages.php'>Messages</a>

<a href='friends.php'>Friends</a>

<a href='members.php'>Members</a>

<a href='logout.php'>Log out</a>

</div>

</header>

<script src="assets/js/jquery.min.js"></script>

            <script src="assets/js/skel.min.js"></script>

            <script src="assets/js/util.js"></script>

            <script src="assets/js/main.js"></script>

<?php

} else {

}

?>

**Home.php**

<?php

require\_once 'header.php';

if (!$loggedin) die("You have been logged out <a href='index.php'>click here </a> to login");

echo "<section><div class='main'><h3> Welcome ". $user."!</h3></section>";

?>

<html>

    <head>

        <title>Grad-Bevy</title>

        <meta charset="utf-8" />

        <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no" />

        <link rel="stylesheet" href="assets/css/main.css" />

</head>

<body>

    <section id="banner">

            <div class="inner">

                <h1>Grad-Bevy:<span>COLLEGE SOCIAL NETWORK </br></span></h1>

                <h2>Great Stories Begin Here!</h2>

                <h3><strong>Think, Explore and Meet your peers</strong></h3>

            </div>

</section>

        <section id="bannera">

                <header>

                    <h2>Academic Calender</h2>

                </header>

                <ul class="actions">

                    <li><a href="academiccal.pdf" class="button alt"><font color="#330033">Click here</font></a><strong>to view the academic calender</strong></li>

                </ul>

        </section>

**Friends.php**

<?php // Example 26-10: friends.php

require\_once 'header.php';

if (!$loggedin) die("You have been logged out <a href='index.php'>click here </a> to login");

if (isset($\_GET['view'])) $view = sanitizeString($\_GET['view']);

else $view = $user;

if ($view == $user) {

$name1 = $name2 = "Your";

$name3 = "You are";

} else {

$name1 = "<a href='members.php?view=$view'>$view</a>'s";

$name2 = "$view's";

$name3 = "$view is";

}

echo "<div class='main'>";

$followers = array();

$following = array();

$result = queryMysql("SELECT \* FROM friends WHERE user='$view'");

$num = $result->num\_rows;

for ($j = 0; $j < $num; ++$j) {

$row = $result->fetch\_array(MYSQLI\_ASSOC);

$followers[$j] = $row['friend'];

}

$result = queryMysql("SELECT \* FROM friends WHERE friend='$view'");

$num = $result->num\_rows;

for ($j = 0; $j < $num; ++$j) {

$row = $result->fetch\_array(MYSQLI\_ASSOC);

$following[$j] = $row['user'];

}

$mutual = array\_intersect($followers, $following);

$followers = array\_diff($followers, $mutual);

$following = array\_diff($following, $mutual);

$friends = FALSE;

if (sizeof($mutual)) {

echo "<span class='subhead'>$name2 mutual friends</span><ul>";

foreach ($mutual as $friend)

echo "<li><a href='members.php?view=$friend'>$friend</a>";

echo "</ul>";

$friends = TRUE;

}

if (sizeof($followers)) {

echo "<span class='subhead'>$name2 followers</span><ul>";

foreach ($followers as $friend)

echo "<li><a href='members.php?view=$friend'>$friend</a>";

echo "</ul>";

$friends = TRUE;

}

if (sizeof($following)) {

echo "<span class='subhead'>$name3 following</span><ul>";

foreach ($following as $friend)

echo "<li><a href='members.php?view=$friend'>$friend</a>";

echo "</ul>";

$friends = TRUE;

}

if (!$friends) echo "<br>You don't have any friends yet.<br><br>";

echo "<a class='button' href='messages.php?view=$view'>"

"View $name2 messages</a>";

?>

</div><br>

</body>

</html>

**Profile.php**

<?php

require\_once 'header.php';

if (!$loggedin) die("You have been logged out <a href='index.php'>click here </a> to login");

echo "<div class='main'><h3> ". $user."'s profile!</h3>";

$result = queryMysql("SELECT \* FROM profiles WHERE user='$user'");

if (isset($\_POST['text'])) {

$text = sanitizeString($\_POST['text']);

$text = preg\_replace('/\s\s+/', ' ', $text);

if ($result->num\_rows)

queryMysql("UPDATE profiles SET text='$text' where user='$user'");

else queryMysql("INSERT INTO profiles VALUES('$user', '$text')");

} else {

if ($result->num\_rows) {

$row = $result->fetch\_array(MYSQLI\_ASSOC);

$text = stripslashes($row['text']);

} else $text = "";

}

$text = stripslashes(preg\_replace('/\s\s+/', ' ', $text));

if (isset($\_FILES['image']['name'])) {

$saveto = "$user.jpg";

move\_uploaded\_file($\_FILES['image']['tmp\_name'], $saveto);

$typeok = TRUE;

switch ($\_FILES['image']['type']) {

case "image/gif":

$src = imagecreatefromgif($saveto);

break;

case "image/jpeg": // Both regular and progressive jpegs

case "image/pjpeg":

$src = imagecreatefromjpeg($saveto);

break;

case "image/png":

$src = imagecreatefrompng($saveto);

break;

default:

$typeok = FALSE;

break;

}

if ($typeok) {

list($w, $h) = getimagesize($saveto);

$max = 100;

$tw = $w;

$th = $h;

if ($w > $h && $max < $w) {

$th = $max / $w \* $h;

$tw = $max;

} elseif ($h > $w && $max < $h) {

$tw = $max / $h \* $w;

$th = $max;

} elseif ($max < $w) {

$tw = $th = $max;

}

$tmp = imagecreatetruecolor($tw, $th);

imagecopyresampled($tmp, $src, 0, 0, 0, 0, $tw, $th, $w, $h);

imageconvolution($tmp, array(array(-1, -1, -1),

array(-1, 16, -1), array(-1, -1, -1)), 8, 0);

imagejpeg($tmp, $saveto);

imagedestroy($tmp);

imagedestroy($src);

}

}

showProfile($user);

echo <<<\_END

<form method='post' action='profile.php' enctype='multipart/form-data'>

<h3>Enter or edit your details and/or upload an image</h3>

<textarea name='text' cols='50' rows='3'>$text</textarea><br>

\_END;

?>

Image: <input type='file' name='image' size='14'>

<input type='submit' value='Save Profile'>

</form></div><br>

</body>

</html>

**Messages.php**

<?php

require\_once 'header.php';

if (!$loggedin) die("You have been logged out <a href='index.php'>click here </a> to login");

if (isset($\_GET['view'])) $view = sanitizeString($\_GET['view']);

else $view = $user;

if (isset($\_POST['text'])) {

$text = sanitizeString($\_POST['text']);

if ($text != "") {

$pm = substr(sanitizeString($\_POST['pm']), 0, 1);

$time = time();

queryMysql("INSERT INTO messages VALUES(NULL, '$user',

'$view', '$pm', $time, '$text')");

}

}

if ($view != "") {

if ($view == $user) $name1 = $name2 = "Your";

else {

$name1 = "<a href='members.php?view=$view'>$view</a>'s";

$name2 = "$view's";

}

echo "<div class='main'><h3>$name1 Messages</h3>";

echo <<<\_END

<form method='post' action='messages.php?view=$view'>

Type here to leave a message:<br>

<textarea name='text' cols='40' rows='3'></textarea><br>

<input type='radio' name='pm' value='0' checked='checked'>

<input type='radio' name='pm' value='1'>

<input type='submit' value='Post Message'></form><br>

\_END;

$query = "SELECT \* FROM messages WHERE recip='$view' ORDER BY time DESC";

$result = queryMysql($query);

$num = $result->num\_rows;

for ($j = 0; $j < $num; ++$j) {

$row = $result->fetch\_array(MYSQLI\_ASSOC);

if ($row['pm'] == 0 || $row['auth'] == $user || $row['recip'] == $user) {

echo date('M jS \'y g:ia:', $row['time']);

echo " <a href='messages.php?view=" . $row['auth'] . "'>" . $row['auth'] . "</a> ";

if ($row['pm'] == 0)

echo "wrote: &quot;" . $row['message'] . "&quot; ";

else

echo "whispered: <span class='whisper'>&quot;" .

$row['message'] . "&quot;</span> ";

}

}

}

if (!$num) echo "<br><span class='info'>No messages yet</span><br><br>";

echo "<br><a class='button' href='messages.php?view=$view'>Refresh messages</a>";

?>

</div><br>

</body>

</html>

# CHAPTER 6

## TESTING

This chapter gives the outline of the testing methods that are carried out to get a bug free system. Quality can be achieved by testing the product using different techniques at different phases of the project development. The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components sub-assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

### 6.1 TESTING PROCESS

Testing is an integral part of software development. Testing process certifies whether the product that is developed compiles with the standards that it was designed to. Testing process involves building of test cases against which the product has to be tested.

### 6.2 TESTING OBJECTIVES

The main objectives of testing process are as follows:

* Testing is a process of executing a program with the intent of finding an error.
* A good test case is one that has high probability of finding undiscovered error.
* A successful test is one that uncovers the undiscovered error.

**Table 6.1: Test Cases**

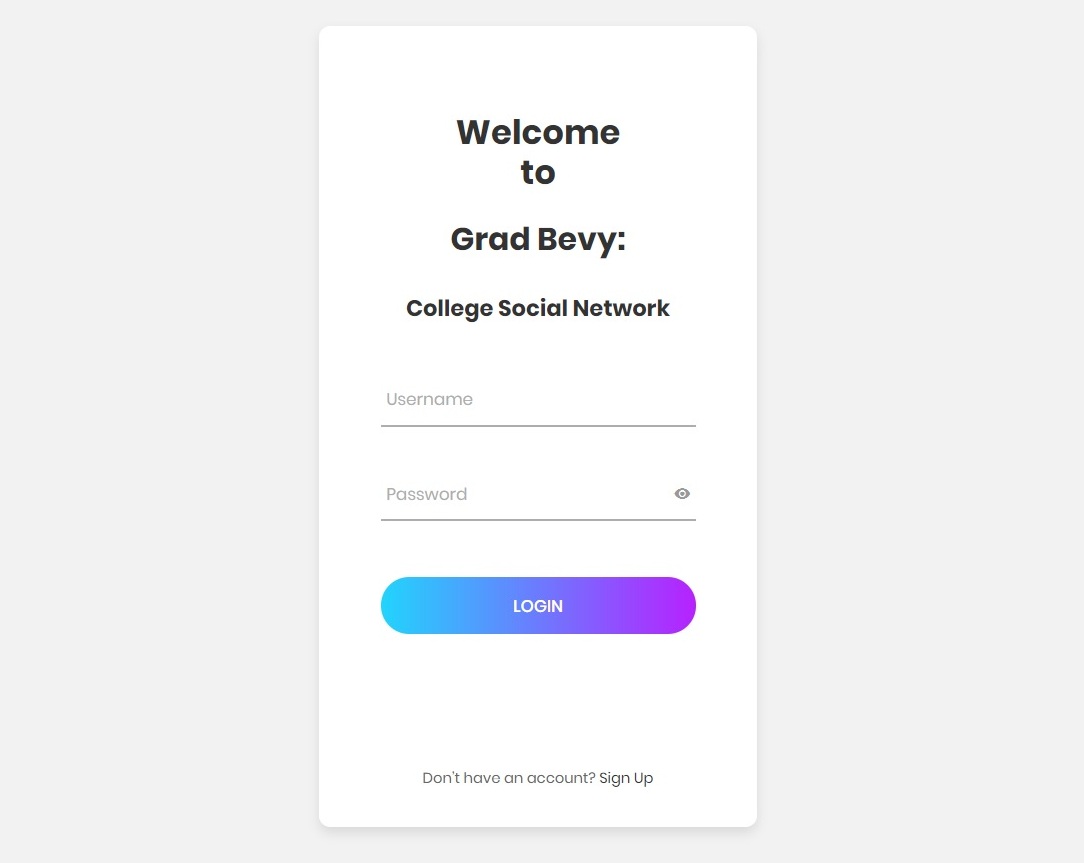
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.NO** | **CASE** | **INPUT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** |
| 1 | User Registration | Blank Field | Submission unsuccessful | Submission unsuccessful |
| 2 | User Registration | Name, Username & Password | Submission successful | Submission successful |
| 3 | User Login | Blank Field | Please fill out the fields | Please fill out the fields |
| 4 | User Login | Username and Password | Home Page | Home Page |
| 5 | User Profile | Blank Field | Cannot be empty | Cannot be empty |
| 6 | User Profile | User’s Bio | Successfully Posted | Successfully Posted |
| 7 | Messages | Blank Field | No messages yet | No messages yet |
| 8 | Messages | Text ‘Hello’ | User wrote ‘Hello’ | User wrote ‘Hello’ |

# CHAPTER 7

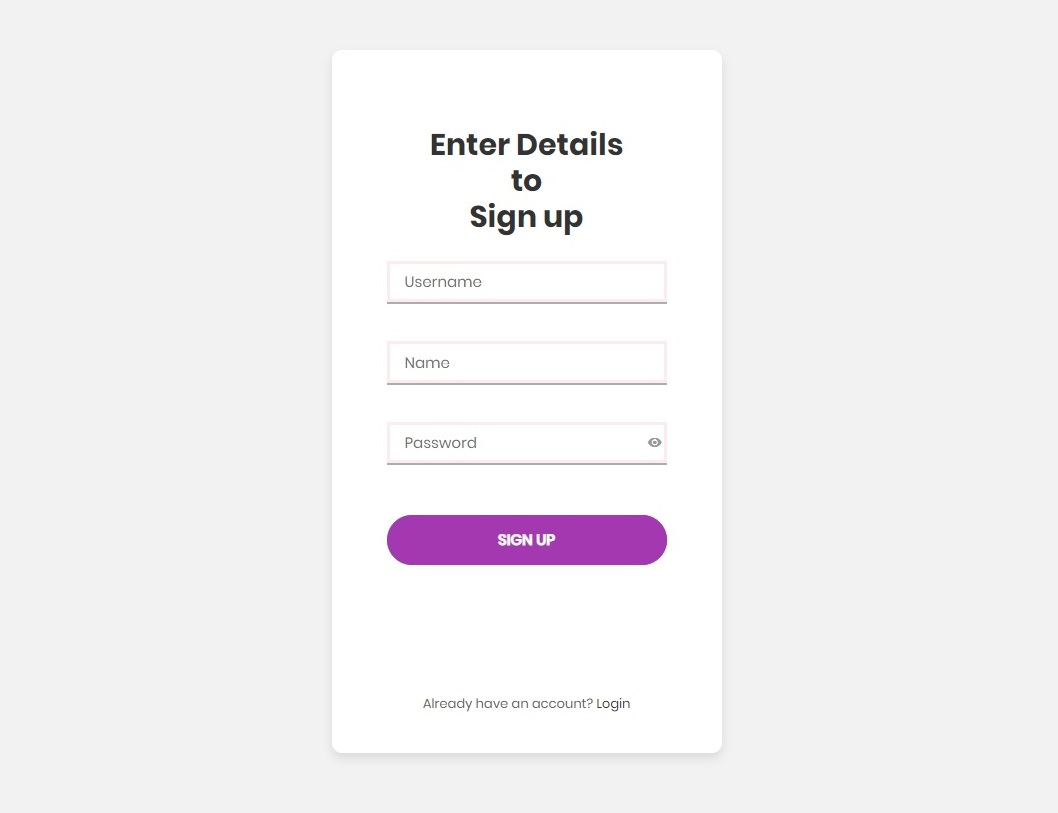
## RESULTS

This section describes the screens of the “College Social Network”.

The snapshots are shown below for each module.



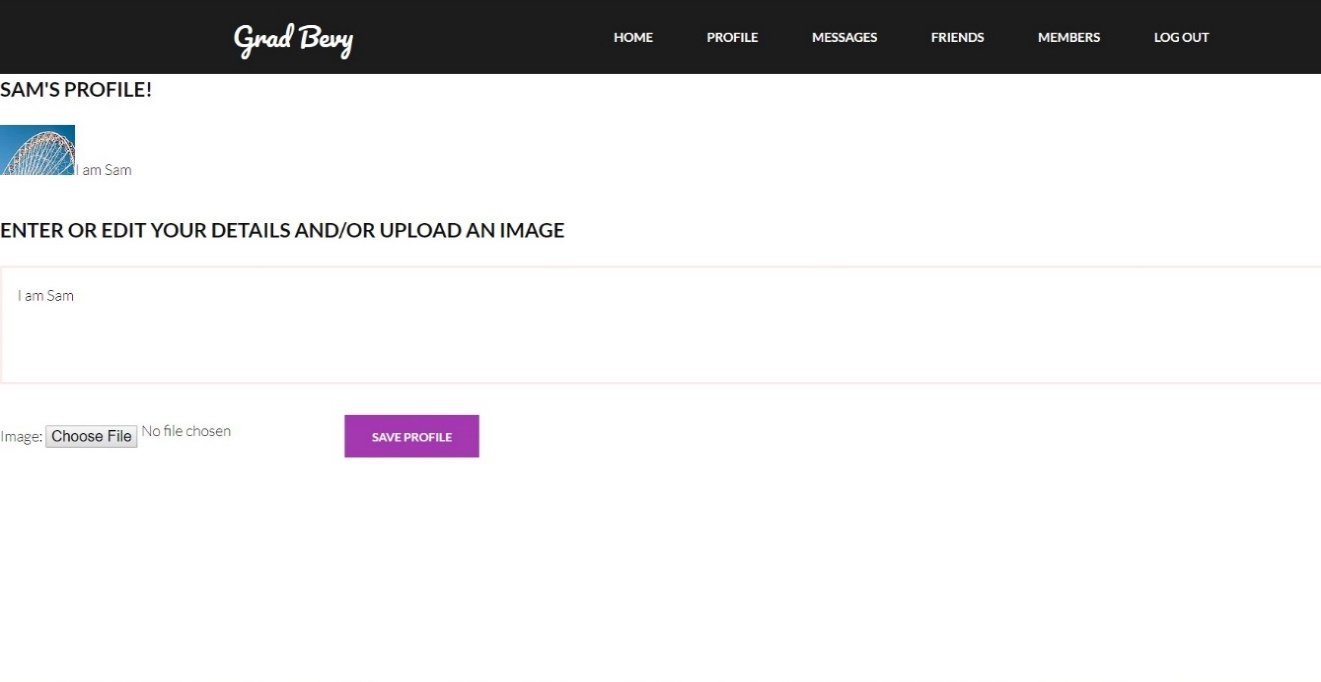
### Figure 7.1: Login Page



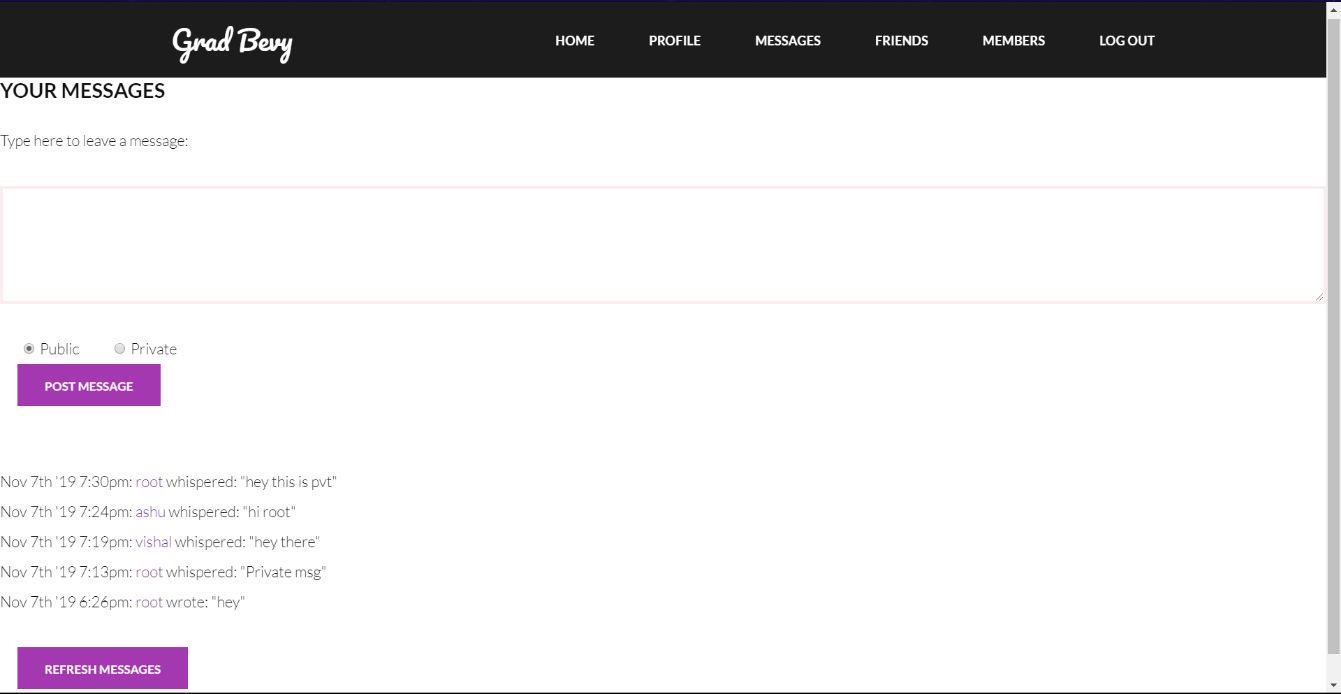
### Figure 7.2: Sign-up Page



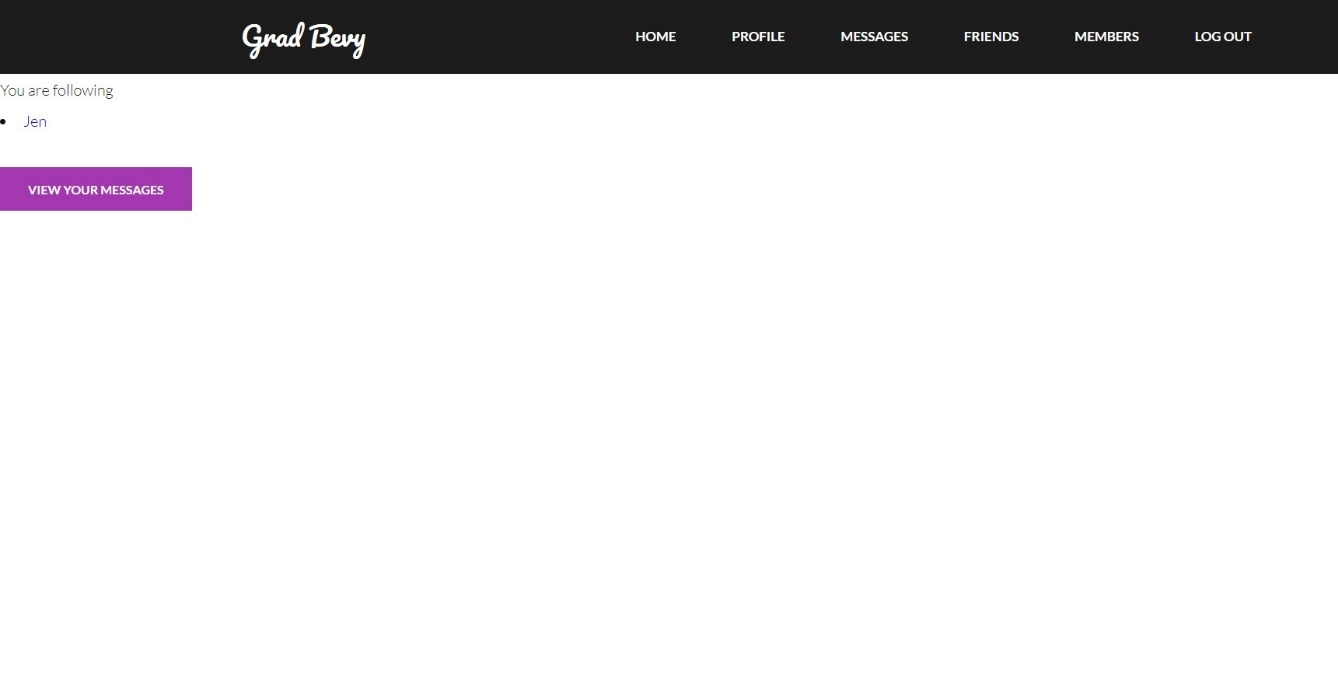
**Figure 7.3: Home Page**



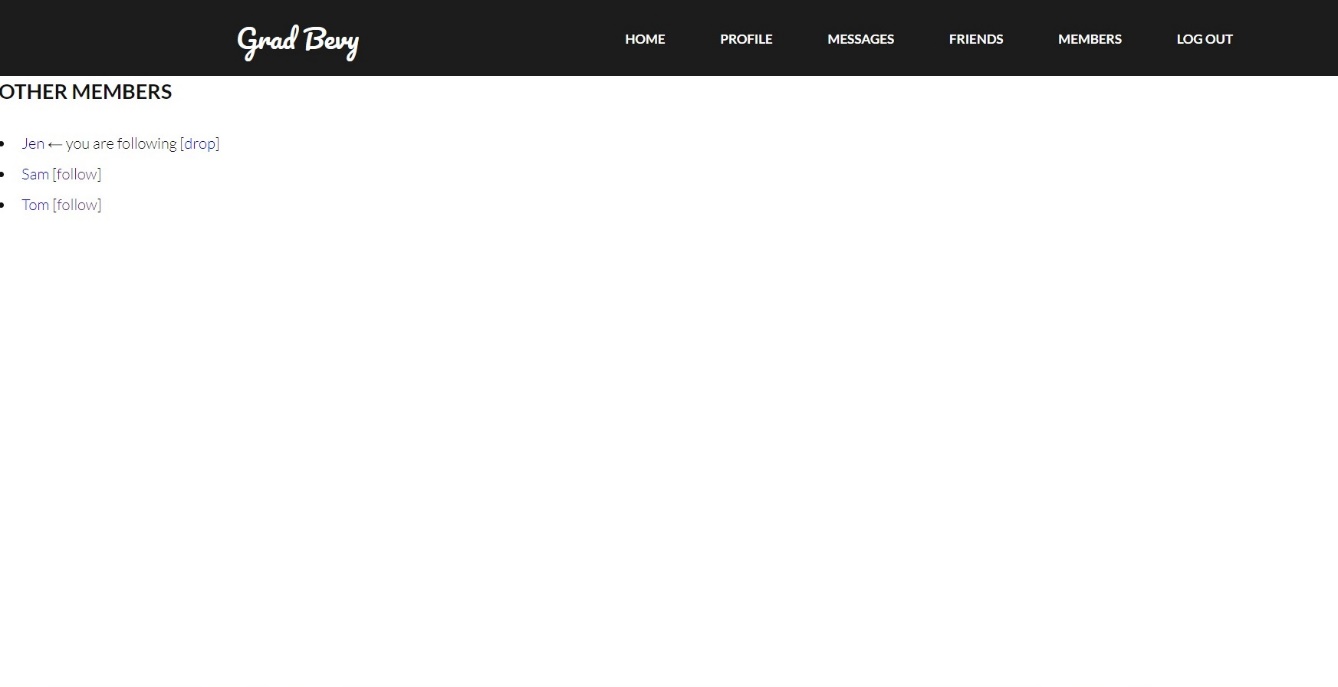
### Figure 7.4: User Profile Page



### Figure 7.5: Messages Page



### Figure 7.6: Friends Page



### Figure 7.7: Members Page

## CONCLUSION

With the theoretical inclination of our syllabus it becomes very essential to take the at most advantage of any opportunity of gaining practical experience that comes along. The building blocks of this project “COLLEGE SOCIAL NETWORK” was one of these opportunities. It gave us the requisite practical knowledge to supplement the already taught theoretical concepts thus making us more competent as a computer engineer. The project from a personal point of view also helped us in understanding the following aspects of project development:

* The planning that goes into implementing a project.
* The importance of proper planning and an organized methodology.
* The key element of team spirit and co-ordination in a successful project.

## BIBILIOGRAPHY

[1]. Randy Connolly, Ricardo Hoar, **“Fundamentals of Web Development”**, 1st Edition, Pearson Education India.

### [2]. Robin Nixon, “Learning PHP, MySQL & JavaScript with jQuery, CSS and HTML5”, 4th Edition, O’Reilly Publications, 2015.

[3]. <https://www.w3schools.com/>

[4]. <https://github.com/>

[5]. <https://stackoverflow.com/>