|  |  |  |
| --- | --- | --- |
| **VISVESVARAYA TECHNOLOGICAL UNIVERSITY** | | |
| **“JNANA SANGAMA”, BELGAUM-590014, KARNATAKA** | | |
| **A Mini Project Report** | | |
|  | **On** |  |
| **“SOCIOGRAM”** | | |
| Submitted in partial fulfillment of the requirement for the Ⅶ semester course of | | |
| **BACHELOR OF ENGINEERING** | | |
|  | In |  |
| **COMPUTER SCIENCE AND ENGINEERING** | | |
| Submitted By | | |
| **SANJAY P D** | **&** | **VAIBHAV SHASTRI** |
| **USN: 1AP17CS031** |  | **USN: 1AP17CS033** |
| Under the guidance of | | |
| **Mr. Muniraju M** |  | **Mrs. Rajashree M Byalal** |
| Assistant Professor, |  | Assistant Professor, |
| Dept of CS & E |  | Dept of CS & E |
| **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING** | | |
| **APS COLLEGE OF ENGINEERING** | | |
| **Anantha Gnana Gangothri** | | |
| **26thk.m, N.H-209, Kanakapura Road, Bangalore–560 082** | | |
| **2019–2020** | | |





# APS COLLEGE OF ENGINEERING

### Anantha Gnana Gangothri,

**NH-209, Kanakapura Road, Bangalore–560 082**

#### DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

**CERTIFICATE**

This is to certify that the project work entitled

**“SOCIOGRAM”**

is a bonafide work carried out by

|  |
| --- |
| **SANJAY P D & VAIBHAV SHASTRI** |
| **1AP17CS031 1AP17CS033** |

In partial fulfillment of the requirement for **“Web Laboratory with Mini Project”** of seventh semester Bachelor of Engineering in Computer Science & Engineering of Visvesvaraya Technological University, Belgaum during the year 2020-2021.

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the **“Web Laboratory with Mini Project”** of seventh semester Bachelor of Engineering in Computer Science and Engineering.

|  |  |  |
| --- | --- | --- |
| **-------------------------------** | **-----------------------------------------** | **---------------------------------** |
| **Mr. Muniraju M, MTECH** | **Mrs. Rajashree M Byalal, MTECH** | **Dr. Bhagyashree R, MTECH** |
| Assistant Professor, | Assistant Professor | Assistant Professor & Head, |
| Dept. of CS & E, | Dept. of CS & E, | Dept. of CS & E, |
| **APSCE** | **APSCE** | **APSCE** |
| **Name of the External**  **1.** |  | **Signature** |
| **2.** |  |  |
|  |  |  |
|  |  |  |

# ACKNOWLEDGEMENT

I take immense pleasure in thanking **Dr. B M Satish**, Principal of APSCE for having permitted us to carry out our project on **“Sociogram”**.

I wish to express our deep sense of gratitude to **Mrs. Bhagyashree R**, Assistant Professor and Head, Dept of CS & E, for her able guidance and useful suggestions, which helped us in completing this project.

I also express our deep sense of gratitude to our guides **Mr. Muniraju M**, Assistant Professor, Dept of CS & E and **Mrs. Rajashree M Byalal**, Assistant Professor, Dept of CS & E for their timely assistance which helped us to complete our project.

Words are inadequate in offering our thanks to all the teaching and non-teaching staff of Dept of CS & E for their encouragement and co-operation in carrying out our project.

Finally, yet importantly, we would like to express our heartfelt thanks to our beloved parents for their blessings, our friends for their help and wishes for successful completion of this project.

|  |
| --- |
| **SANJAY P D & VAIBHAV SHASTRI** |
| **1AP17CS031 1AP17CS033** |

|  |  |  |
| --- | --- | --- |
| **SERIAL NO.** | **CHAPTER** | **PAGE NO.** |
| **1** | **INTRODUCTION** | **1** |
| 1.1 | History of Web Technology | 1 |
| 1.2 | Introduction to Web Technology | 3 |
| 1.2.1 | Advantages of Web Technology | 3 |
| 1.2.2 | Disadvantages of Web Technology | 3 |
| **2** | **REQUIREMENT ANALYSIS** | **4** |
| 2.1 | Web Technologies | 4 |
| 2.1.1 | HTML | 4 |
| 2.1.2 | CSS | 5 |
| 2.1.3 | BOOTSTRAP | 5 |
| 2.2 | Scripting Language | 6 |
| 2.2.1 | JAVASCRIPT | 6 |
| 2.2.2 | PHP | 6 |
| 2.2.3 | MYSQL | 7 |
| 2.3 | Web platform | 7 |
| 2.3.1 | APACHE Server | 7 |
| **3** | **REQIREMENTS SPECIFICATION** | **7** |
| 3.1 | Hardware Requirements | 7 |
| 3.2 | Software Requirements | 7 |
| **4** | **ANALYSIS AND DESIGN** | **8** |
| 4.1 | Analysis | 8 |
| 4.1.1 | Website Development Life Cycle | 9 |
| 4.2 | Design | 10 |
| 4.2.1 | Physical Design | 10 |
| 4.2.2 | Database Design | 10 |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **5** | **IMPLEMENTATION** | **11** |
| 5.1 | Source Code | 11 |
| **6** | **SCREENSHOTS** | **18** |
| **7** | **TESTING** | **22** |
| 7.1 | Testing Techniques | 22 |
| 7.2 | Software Testing Fundamentals | 22 |
| 7.3 | Testing Objective | 22 |
| 7.4 | Test Cases | 23 |
| **8** | **RESULT AND CONCLUSION** | **24** |
| **9** | **BIBLIOGRAPHY** | **25** |
|  |  |  |
|  | **TABLE OF FIGURES** |  |
| **FIG NO** | **TITLE** | **PAGE NO.** |
| 1.1 | Tim’s Original proposal | 2 |
| 4.1.1.1 | Web System Architecture | 9 |
| 6.1 | Login page of the website | 18 |
| 6.2 | Registration page | 18 |
| 6.3 | Sociogram home page | 19 |
| 6.4 | Friend requests page | 19 |
| 6.5 | Friend section | 20 |
| 6.6 | Search page | 20 |
| 6.7 | Profile page | 21 |
|  | **TABLES** |  |
| 7.4 | TEST CASES | 23 |

## ABSTRACT

Web programming, also known as web development, is the creation of dynamic web applications. Web development is the work involved in developing a web site for the Internet (World Wide Web) or an intranet (a private network). A more comprehensive list of tasks to which web development commonly refers, may include web engineering, web design, web content development, etc. There are two broad divisions of web development – front-end development (also called client- side development) and back-end development (also called server-side development). Front-end development refers to constructing what a user sees when they load a web application – the content, design and how you interact with it. This is done with three codes – HTML, CSS and JavaScript. Back-end development controls what goes on behind the scenes of a web application. A back-end often uses a database to generate the front-end. Web development takes into account many security considerations, such as data entry error checking through forms, filtering output, and encryption.

An ever-growing set of tools and technologies help us build more dynamic and interactive websites. Further, web developers now help to deliver applications as web services which were traditionally only available as applications on a desk- based computer.

**USAGES:**

* It is used to view images and other texting contents.
* It is used for communication purpose.
* It is used to create a social image of a person or content.
* Users can text and send posts to other users.
* Users can update profile after logging-in to their respective credentials.
* Users can send and accept requests to connect.
* Users can make their posts public or private.
* Users can create their multiple accounts.

**CHAPTER 1**

**INTRODUCTION**

* 1. **History of Web Technology:**

Sir Tim Berners-Lee is a British computer scientist. He was born in London, and his parents were early computer scientists, working on one of the earliest computers.

Growing up, Sir Tim was interested in trains and had a model railway in his bedroom. He [recalls](http://www.w3.org/People/Berners-Lee/Kids.html):

“I made some electronic gadgets to control the trains. Then I ended up getting more interested in electronics than trains. Later on, when I was in college, I made a computer out of an old television set.”

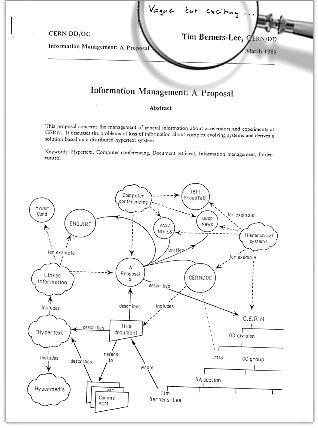
After graduating from Oxford University, Berners-Lee became a software engineer at [CERN](http://home.web.cern.ch/), the large particle physics laboratory near Geneva, Switzerland. Scientists come from all over the world to use its accelerators, but Sir Tim noticed that they were having difficulty sharing information.

“In those days, there was different information on different computers, but you had to log on to different computers to get at it. Also, sometimes you had to learn a different program on each computer. Often it was just easier to go and ask people when they were having coffee…”, [Tim](http://www.w3.org/People/Berners-Lee/Kids.html) [says.](http://www.w3.org/People/Berners-Lee/Kids.html)

Tim thought he saw a way to solve this problem – one that he could see could also have much broader applications. Already, millions of computers were being connected together through the fast-deve[loping internet](http://www.internetsociety.org/internet/what-internet) and Berners-Lee realized they could share information by exploiting an emerging technology called hypertext.

In March 1989, Tim laid out his vision for what would become the web in a document called “[Information Management: A Proposal](http://info.cern.ch/Proposal.html)”. Believe it or not, Tim’s initial proposal was not immediately accepted. In fact, his boss at the time, [Mike Sendall](http://bullarchive.web.cern.ch/bullarchive/9930/art2/Text_E.html), noted the words “Vague but exciting” on the cover.

The web was never an official CERN project, but Mike managed to give Tim time to work on it in September 1990. He began work using a [NeXT computer,](http://en.wikipedia.org/wiki/NeXT_Computer) one of Steve Jobs’ early products.



**Figure 1.1 Tim’s original proposal. Image: CERN**

By October of 1990, Tim had written the three fundamental technologies that remain the foundation of today’s web (and which you may have seen appear on parts of your web browser):

* + - HTML: Hyper Text Mark-up Language. The mark-up (formatting) language for the web.
    - URI: Uniform Resource Identifier. A kind of “address” that is unique and used to identify to each resource on the web. It is also commonly called a URL.
    - HTTP: Hypertext Transfer Protocol. Allows for the retrieval of linked resources from across the web.

Tim also wrote the first web page editor/browser (“WorldWideWeb.app”) and the first web server (“http”). By the end of 1990, the first web page was served on the open internet, and in 1991, people outside of CERN were invited to join this new web community.

## Introduction to Web Technology

Web technology is the development of the mechanism that allows two of more computer devices to communicate over a network. For instance, in a typical office setting, a number of computers plus additional devices such as printers may be interconnected via a network, allowing for quick and convenient transmission of information. The processes involved in web technology are complex and diverse, which is why major businesses employ whole departments to deal with the issue. Web technology has revolutionized communication methods and has made operations far more efficient.

### Advantages of Web Technology

The main advantage of web technology is that it offers convenience and a high speed of communication in the computer world. Whether in the office or the home, processes using a computer are swifter and more straightforward with the use of a network. Web technology allows messages to be sent around a system, whereas before it may have been necessary to employ a runner or leave your workspace to communicate a message. It is clear to see how web technology reduces costs and makes a company more efficient, raising business potential.

### Disadvantages of Web Technology

Matters involving web technology can be very complicated, and it would be difficult for someone without relevant experience to sort a network problem out. This means it is necessary to employ someone with the specific skills to solve network issues, which costs money. Additionally, the existence of a network provides the opportunity for an attack on the computer system. Weaknesses in a network could be exploited; important information could be stolen or destroyed and malware could infect the various network systems. For this reason, network security is another issue that must be considered when using web technology.

## CHAPTER 2

**REQUIREMENT ANALYSIS**

* 1. **WEB TECHNOLOGIES**
     1. **HTML**

Hypertext Mark-up Language (HTML) is the standard mark-up language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

HTML is a mark-up language that web browsers use to interpret and compose text, images, and other material into visual or audible web pages. Default characteristics for every item of HTML mark-up are defined in the browser, and these characteristics can be altered or enhanced by the web page designer's additional use of CSS. Web browsers receive HTML documents from a [web server](https://en.wikipedia.org/wiki/Web_server) or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.

With HTML constructs, [images](https://en.wikipedia.org/wiki/HTML_element#Images_and_objects) and other objects such as [interactive forms](https://en.wikipedia.org/wiki/Fieldset) may be embedded into the rendered page. [HTML elements](https://en.wikipedia.org/wiki/HTML_element) are the building blocks of HTML pages. HTML provides a means to create [structured documents](https://en.wikipedia.org/wiki/Structured_document) by denoting structural [semantics](https://en.wikipedia.org/wiki/Semantics) for text such as headings, paragraphs, lists, [links](https://en.wikipedia.org/wiki/Hyperlink), quotes and other items. HTML elements are delineated by tags, written using [angle brackets.](https://en.wikipedia.org/wiki/Bracket#Angle_brackets) Tags such as <a> and <img /> directly introduce content into the page. Other tags such as <body> with <p> surround and provide information about document text and may include other tags as sub-elements, like lists; <ol>, <ul>, <li>. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML mark-up consists of several key components, including those called tags (and their attributes), character-based data types, character references and entity references. HTML tags most commonly come in pairs like <h1> and </h1>, although some represent empty elements and so are unpaired, for example <img>. The first tag in such a pair is the start tag, and the second is the end tag (they are also called opening tags and closing tags).

### CSS

**Cascading Style Sheets** (**CSS**) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media. The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

CSS (Cascading Style Sheets) is used to style and lay out web pages - for example, to alter the font, color, size, and spacing of your content, split it into multiple columns, or add animations and other decorative features.

Separation of formatting and content also makes it feasible to present the same mark-up page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or [screen reader](https://en.wikipedia.org/wiki/Screen_reader)), and on [Braille-based](https://en.wikipedia.org/wiki/Braille_display) tactile devices. CSS also has rules for alternate formatting if the content is accessed on a [mobile device.](https://en.wikipedia.org/wiki/Mobile_device)

CSS has a simple syntax and uses a number of English keywords to specify the names of various style properties. A style sheet consists of a list of rules. Each rule or rule-set consists of one or more selectors, and a declaration block.

Before CSS, nearly all presentational attributes of HTML documents were contained within the HTML mark-up. All font colors, background styles, element alignments, borders and sizes had to be explicitly described, often repeatedly, within the HTML. CSS lets authors move much of that information to another file, the style sheet, resulting in considerably simpler HTML.

### 2.1.3 BOOTSTRAP

Bootstrap is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first websites. Bootstrap is a free and open-source CSS framework. A precompiled version of Bootstrap is available in the form of one CSS file and three JavaScript files that can be readily added to any project.

Bootstrap is a web framework that focuses on simplifying the development of informative web pages (as opposed to web apps). The primary purpose of adding it to a web project is to apply Bootstrap's choices of color, size, font and layout to that project. As such, the primary factor is whether the developers in charge find those choices to their liking. Once added to the project,

Bootstrap provides basic style definitions for all HTML elements. The result is a uniform appearance for prose, tables and form elements across web browsers. In addition, developers can take advantage of CSS classes defined in Bootstrap to further customize the appearance of their contents. For example, Bootstrap has provisioned for light- and dark-colored tables, page headings, more prominent pull quotes, and text with a highlight.

**2.2 SCRIPTING LANGUAGE**

## 2.2.1 JAVASCRIPT

JavaScript often abbreviated as JS, is a [high-level,](https://en.wikipedia.org/wiki/High-level_programming_language) [interpreted](https://en.wikipedia.org/wiki/Interpreted_language) [programming language.](https://en.wikipedia.org/wiki/Programming_language) It is a language which is also characterized as [dynamic,](https://en.wikipedia.org/wiki/Dynamic_programming_language) [weakly typed,](https://en.wikipedia.org/wiki/Weak_typing) [prototype-based](https://en.wikipedia.org/wiki/Prototype-based_programming) and [multi-](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language) [paradigm.](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language)

Alongside [HTML](https://en.wikipedia.org/wiki/HTML) and [CSS,](https://en.wikipedia.org/wiki/CSS) JavaScript is one of the three core technologies of the [World Wide](https://en.wikipedia.org/wiki/World_Wide_Web) [Web.](https://en.wikipedia.org/wiki/World_Wide_Web) JavaScript enables interactive [web pages](https://en.wikipedia.org/wiki/Web_page) and thus is an essential part of [web](https://en.wikipedia.org/wiki/Web_application) [applications.](https://en.wikipedia.org/wiki/Web_application) The vast majority of [websites](https://en.wikipedia.org/wiki/Website) use it, and all major [web browsers](https://en.wikipedia.org/wiki/Web_browser) have a dedicated [JavaScript engine](https://en.wikipedia.org/wiki/JavaScript_engine) to execute it.

**2.2.2 PHP**

PHP code is usually processed on A web server by a PHP interpreter implemented as a module, a daemon or as a common gateway interface (CGI) executable. On a web server, the result of the interpreted and executed PHP code – which may be any type of data, such as generated HTML or binary image data – would form the whole or part of an HTTP response. Various web template systems, web content management systems, and web frameworks exist which can be employed to orchestrate or facilitate the generation of that response. Additionally, PHP can be used for many programming tasks outside of the web context, such as standalone graphical applications and robotic drone control. Arbitrary PHP code can also be interpreted and executed via command-line interface (CLI).

The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge.

**2.2.3 MySQL**

MySQL is free and open-source software under the terms of the GNU General Public License, and is also available under a variety of proprietary licenses. MySQL was owned and sponsored by the Swedish company MySQL AB, which was bought by Sun Microsystems (now Oracle Corporation). In 2010, when Oracle acquired Sun, Widenius forked the open-source MySQL project to create MariaDB.

MySQL has stand-alone clients that allow users to interact directly with a MySQL database using SQL, but more often MySQL is used with other programs to implement applications that need relational database capability. MySQL is a component of the LAMP web application software stack (and others), which is an acronym for *Linux, Apache, MySQL, Perl/PHP/Python*. MySQL is used by many database-driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Facebook, Flickr,[]](https://en.wikipedia.org/wiki/MySQL#cite_note-12) MediaWiki, Twitter and YouTube.

**2.3 WEB PLATFORM**

**2.3.1 APACHE SERVER**

The **Apache HTTP Server**, colloquially called **Apache** is a free and open-source cross-platform web server software, released under the terms of Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation.

The vast majority of Apache HTTP Server instances run on a Linux distribution but current versions also run-on Microsoft Windows, OpenVMS, and a wide variety of Unix-like systems. Past versions also ran on NetWare, OS/2 and other operating systems, including ports to mainframes.

Originally based on the NCSA HTTP server, development of Apache began in early 1995 after work on the NCSA code stalled. Apache played a key role in the initial growth of the World Wide Web, quickly overtaking NCSA HTTP as the dominant HTTP server, and has remained most popular since April 1996. In 2009, it became the first web server software to serve more than 100 million websites. As of April 2020, Net craft estimated that Apache served 29.12% of the million busiest websites, while Nginx served 25.54% according to W3Techs, Apache served 39.5% of the top 10 million sites and Nginx served 31.7%.

**CHAPTER 3**

# REQUIREMENT SPECIFICATION

## Hardware Requirements

|  |  |
| --- | --- |
| Processor | Intel Pentium 5 or more |
| RAM | 4GB or more |
| Keyboard | Standard keyboard |
| Monitor | 1024x768 display resolution with true color |
| Hard Disk Capacity | 512GB or more |

## Software Requirements

|  |  |
| --- | --- |
| Operating System | Windows, OS and Linux |
| Web Technologies | HTML, CSS and Bootstrap |
| Scripting Languages | JavaScript, PHP, MySQL |
| Web browser | Chrome or internet Explorer |
| Source code editor | Notepad |

## CHAPTER 4

**ANALYSIS AND DESIGN**

* 1. **Analysis**

This website has 5 navigation bars where you can send the friend requests through friend request bar and it will be visible to the requested user, you can set your profile picture and also you could upload image. You could also post through the Post navigation bar where it will be visible in the home section. The friends section in the navigation bar will show all friend requests and in the end we could Log out from the page from logout bar.

It also has a search bar where we could search through emails, hometowns, names and posts. Everything posted will get updated in the home section and with these the user could see all the posts. You can make an account and fill all the required details and make an account. You can also make the posts public and private where the private is only visible to friends. We have other options like you could accept or deny friend requests and also you can add phone number to your profile details.

### Website Development Life Cycle

#### Step 1. Gathering Information:

The most important task at this point is to get the clear understanding of your future website purposes, the main goals you wish to get, and the target audience you want to attract to your site. Such kind of a website development questionnaire helps to develop the best strategy for further project management.

#### Step 2. Planning:

At this stage of website development cycle, the developer creates the data that can give to a customer an opportunity to judge how the entire site will look like. On the basis of the information that was gathered together in the previous phase, the sitemap is created.

#### Step 3. Design:

**Website layout** is the result of designer’s work. It can be a graphic sketch or an actual graphic design. The primary function of the layout is to represent the information structure, visualize the content, and demonstrate the basic functional. Layouts contain colors, logos, images and can give a general understanding of the future product.

#### Step 4. Content Writing and Assembly:

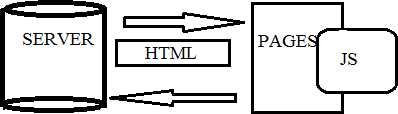
Content writing and compiling usually overlaps with other stages of website creation, and its role can’t be underestimated. At this step it is necessary to put in writing the very essence you’d like to communicate to the audience of your website, and add calls-to-action. Content writing involves also creation of catching headlines, text editing, writing new text, compiling the existing text, etc., which takes time and effort. As a rule, the client undertakes to provide website content ready to migrate to the site. It is better when all website content is provided before or during website coding.

#### Step 5. Coding:

At this step, you can finally start creating the website itself. Graphic elements that have been designed during the previous stages should be used to create an actual website. Usually, the home page is created first, and then all sub-pages are added, according to the website hierarchy.

#### Step 6. Testing:

Testing is probably the most routine and important part of a process. Every single link should be tested to make sure that there are no broken, forms and scripts that are used must be tested before deploying the project, validation of user input must be done.



#### Figure 4.1.1.1 Web System Architecture

* 1. **DESIGN**

The purpose of design phase is to plan a solution for problem specified by the requirements. System design aims to identify the modules that should be in the system, the specification of those modules and how they interact with each other to produce the results. The goal of the design process is to produce a model that can be used later to build that system. The produced model is called design of the system.

System design is the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements. Normally, the design proceeds in two stages: Physical design and Database design.

### Physical Design

The physical design is a graphical representation of a system showing the system’s internal and external entities and the flow of data into and out of these entities. An internal entity is an entity within the system that transforms data.

**4.2.2 Database Design**

**Database design** is the organization of data according to a database model. The designer determines what data must be stored and how the data elements interrelate. With this information, they can begin to fit the data to the database model. Database management system manages the data accordingly.

Database design involves classifying data and identifying interrelationships. This theoretical representation of the data is called an *ontology*. The ontology is the theory behind the database's design.

## CHAPTER 5

**IMPLEMENTATION**

**5.1 Source Code**

<?php

require 'functions/functions.php';

session\_start();

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

header("location:home.php");

}

session\_destroy();

session\_start();

ob\_start();

?>

<!DOCTYPE html>

<html>

<head>

<title>SocioGram Login</title>

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

<style>

.container{

margin: 40px auto;

width: 400px;

}

.content {

padding: 30px;

background-color: white;

box-shadow: 0 0 5px #4267b2;

}

</style>

</head>

<body>

<h1>Welcome to SocioGram</h1>

<div class="container">

<div class="tab">

<button class="tablink active" onclick="openTab(event,'signin')" id="link1">Login</button>

<button class="tablink" onclick="openTab(event,'signup')" id="link2">Sign Up</button>

</div>

<div class="content">

<div class="tabcontent" id="signin">

<form method="post" onsubmit="return validateLogin()">

<label>Email<span>\*</span></label><br>

<input type="text" name="useremail" id="loginuseremail">

<div class="required"></div>

<br>

<label>Password<span>\*</span></label><br>

<input type="password" name="userpass" id="loginuserpass">

<div class="required"></div>

<br><br>

<input type="submit" value="Login" name="login">

</form>

</div>

<div class="tabcontent" id="signup">

<form method="post" onsubmit="return validateRegister()">

<!--Package One-->

<h2>Highly Required Information</h2>

<hr>

<!--First Name-->

<label>First Name<span>\*</span></label><br>

<input type="text" name="userfirstname" id="userfirstname">

<div class="required"></div>

<br>

<!--Last Name-->

<label>Last Name<span>\*</span></label><br>

<input type="text" name="userlastname" id="userlastname">

<div class="required"></div>

<br>

<!--Nickname-->

<label>Nickname</label><br>

<input type="text" name="usernickname" id="usernickname">

<div class="required"></div>

<br>

<!--Password-->

<label>Password<span>\*</span></label><br>

<input type="password" name="userpass" id="userpass">

<div class="required"></div>

<br>

<!--Confirm Password-->

<label>Confirm Password<span>\*</span></label><br>

<input type="password" name="userpassconfirm" id="userpassconfirm">

<div class="required"></div>

<br>

<!--Email-->

<label>Email<span>\*</span></label><br>

<input type="text" name="useremail" id="useremail">

<div class="required"></div>

<br>

<!--Birth Date-->

Birth Date<span>\*</span><br>

<select name="selectday">

<?php

for($i=1; $i<=31; $i++){

echo '<option value="'. $i .'">'. $i .'</option>';

}

?>

</select>

<select name="selectmonth">

<?php

echo '<option value="1">January</option>';

echo '<option value="2">February</option>';

echo '<option value="3">March</option>';

echo '<option value="4">April</option>';

echo '<option value="5">May</option>';

echo '<option value="6">June</option>';

echo '<option value="7">July</option>';

echo '<option value="8">August</option>';

echo '<option value="9">September</option>';

echo '<option value="10">October</option>';

echo '<option value="11">Novemeber</option>';

echo '<option value="12">December</option>';

?>

</select>

<select name="selectyear">

<?php

for($i=2017; $i>=1900; $i--){

if($i == 1996){

echo '<option value="'. $i .'" selected>'. $i .'</option>';

}

echo '<option value="'. $i .'">'. $i .'</option>';

}

?>

</select>

<br><br>

<!--Gender-->

<input type="radio" name="usergender" value="M" id="malegender" class="usergender">

<label>Male</label>

<input type="radio" name="usergender" value="F" id="femalegender" class="usergender">

<label>Female</label>

<div class="required"></div>

<br>

<!--Hometown-->

<label>Hometown</label><br>

<input type="text" name="userhometown" id="userhometown">

<br>

<!--Package Two-->

<h2>Additional Information</h2>

<hr>

<!--Marital Status-->

<input type="radio" name="userstatus" value="S" id="singlestatus">

<label>Single</label>

<input type="radio" name="userstatus" value="E" id="engagedstatus">

<label>Engaged</label>

<input type="radio" name="userstatus" value="M" id="marriedstatus">

<label>Married</label>

<br><br>

<!--About Me-->

<label>About Me</label><br>

<textarea rows="12" name="userabout" id="userabout"></textarea>

<br><br>

<input type="submit" value="Create Account" name="register">

</form>

</div>

</div>

</div>

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

</body>

</html>

<?php

$host = 'localhost';

$username = 'spd';

$password = 'qwerty';

$database = 'webmini';

$conn = mysqli\_connect($host, $username, $password, $database);

if ($\_SERVER['REQUEST\_METHOD'] == 'POST') { // A form is posted

if (isset($\_POST['login'])) { // Login process

$useremail = $\_POST['useremail'];

$userpass = md5($\_POST['userpass']);

$query = mysqli\_query($conn, "SELECT \* FROM users WHERE user\_email = '$useremail' AND user\_password = '$userpass'");

if($query){

if(mysqli\_num\_rows($query) == 1) {

$row = mysqli\_fetch\_assoc($query);

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

$\_SESSION['user\_name'] = $row['user\_firstname'] . " " . $row['user\_lastname'];

echo "VALID";

header("location:home.php");

}

else {

?> <script>

document.getElementsByClassName("required")[0].innerHTML = "Invalid Login Credentials.";

document.getElementsByClassName("required")[1].innerHTML = "Invalid Login Credentials.";

</script> <?php

}

} else{

echo mysqli\_error($conn);

}

}

if (isset($\_POST['register'])) { // Register process

// Retrieve Data

$userfirstname = $\_POST['userfirstname'];

$userlastname = $\_POST['userlastname'];

$usernickname = $\_POST['usernickname'];

$userpassword = md5($\_POST['userpass']);

$useremail = $\_POST['useremail'];

$userbirthdate = $\_POST['selectyear'] . '-' . $\_POST['selectmonth'] . '-' . $\_POST['selectday'];

$usergender = $\_POST['usergender'];

$userhometown = $\_POST['userhometown'];

$userabout = $\_POST['userabout'];

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

$userstatus = $\_POST['userstatus'];

}

else{

$userstatus = NULL;

}

// Check for Some Unique Constraints

$query = mysqli\_query($conn, "SELECT user\_nickname, user\_email FROM users WHERE user\_nickname = '$usernickname' OR user\_email = '$useremail'");

if(mysqli\_num\_rows($query) > 0){

$row = mysqli\_fetch\_assoc($query);

if($usernickname == $row['user\_nickname'] && !empty($usernickname)){

?> <script>

document.getElementsByClassName("required")[4].innerHTML = "This Nickname already exists.";

</script> <?php

}

// Insert Data

$sql = "INSERT INTO users(user\_firstname, user\_lastname, user\_nickname, user\_password, user\_email, user\_gender, user\_birthdate, user\_status, user\_about, user\_hometown)

VALUES ('$userfirstname', '$userlastname', '$usernickname', '$userpassword', '$useremail', '$usergender', '$userbirthdate', '$userstatus', '$userabout', '$userhometown')";

$query = mysqli\_query($conn, $sql);

if($query){

$query = mysqli\_query($conn, "SELECT user\_id FROM users WHERE user\_email = '$useremail'");

$row = mysqli\_fetch\_assoc($query);

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

header("location:home.php");

}

}

} ?>

**NOTE:** The above source code is applicable just for only one section from this project. For other sections the source code can be generated. But we just took an example of one section (index page) and explained the source code from this Project.

## CHAPTER 6

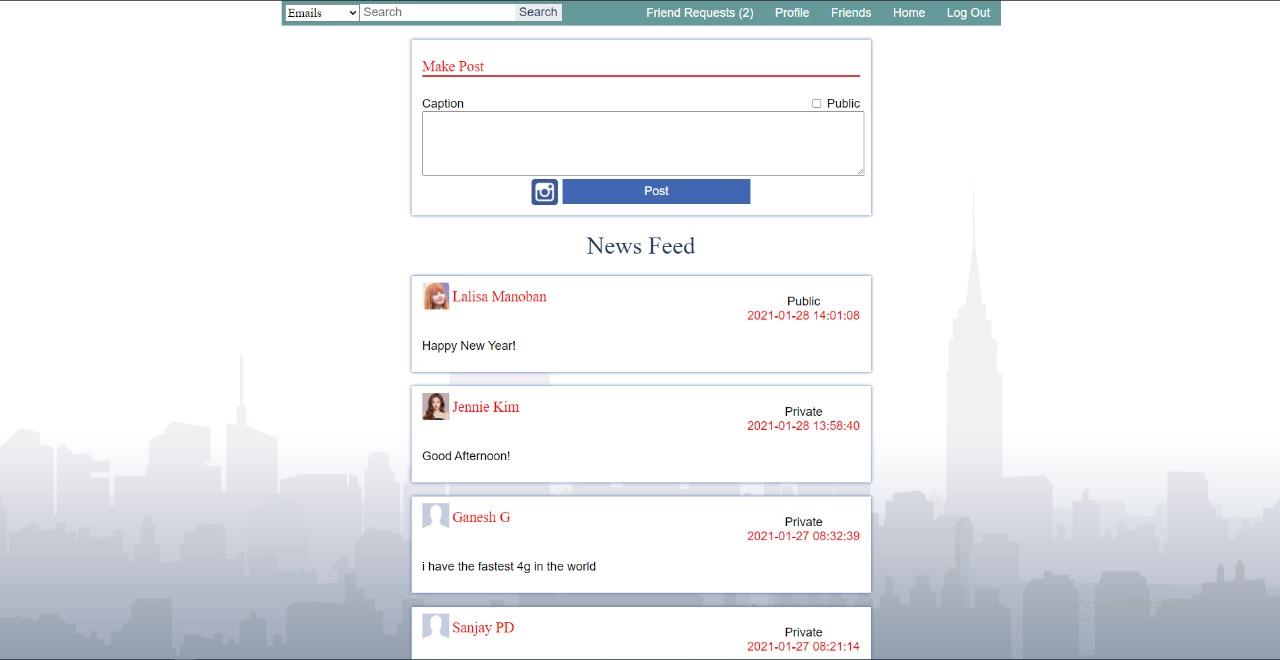
**SCREENSHOTS**

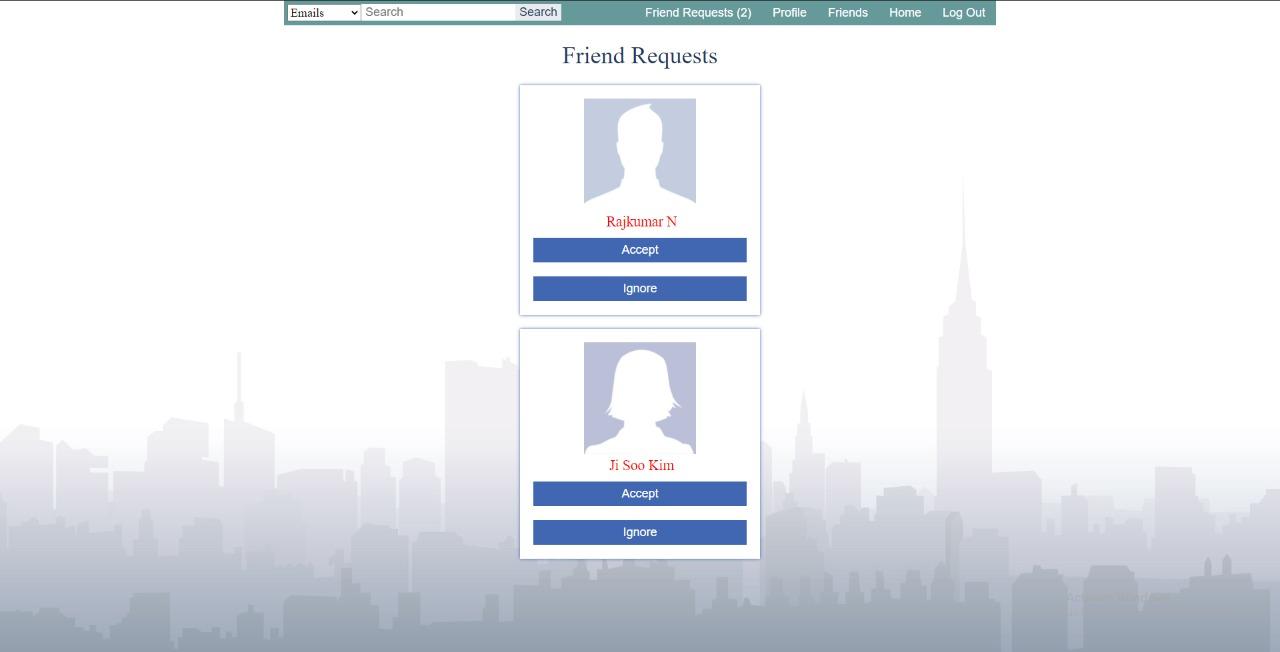
****

#### Figure 6.1 Login Page of the website

#### 

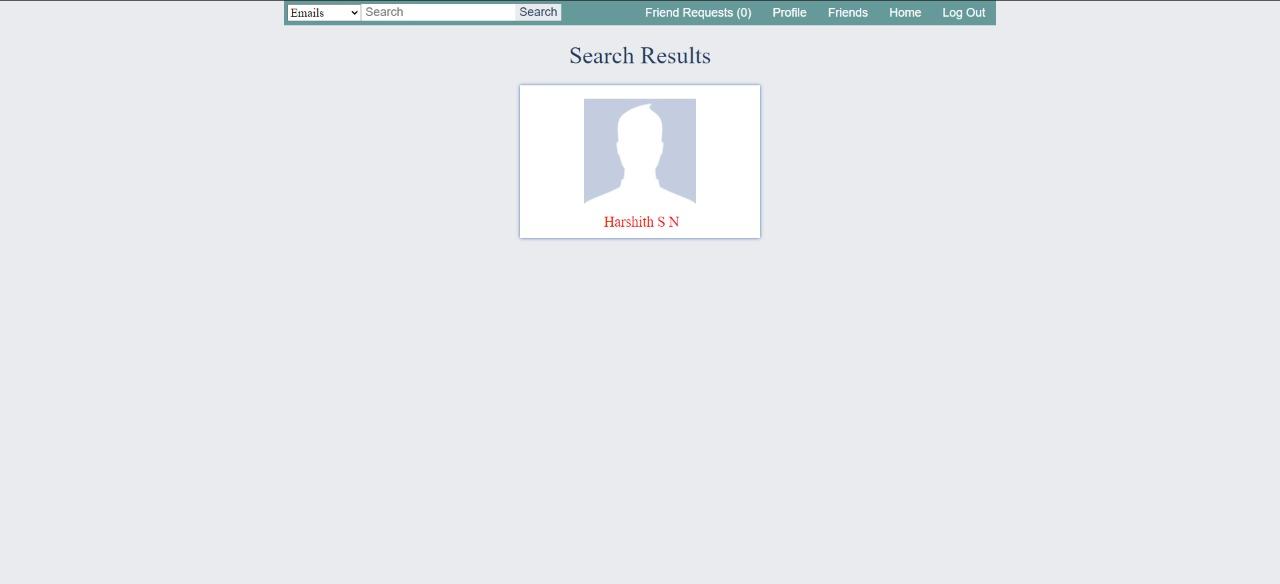
#### Figure 6.2 Registration page

****

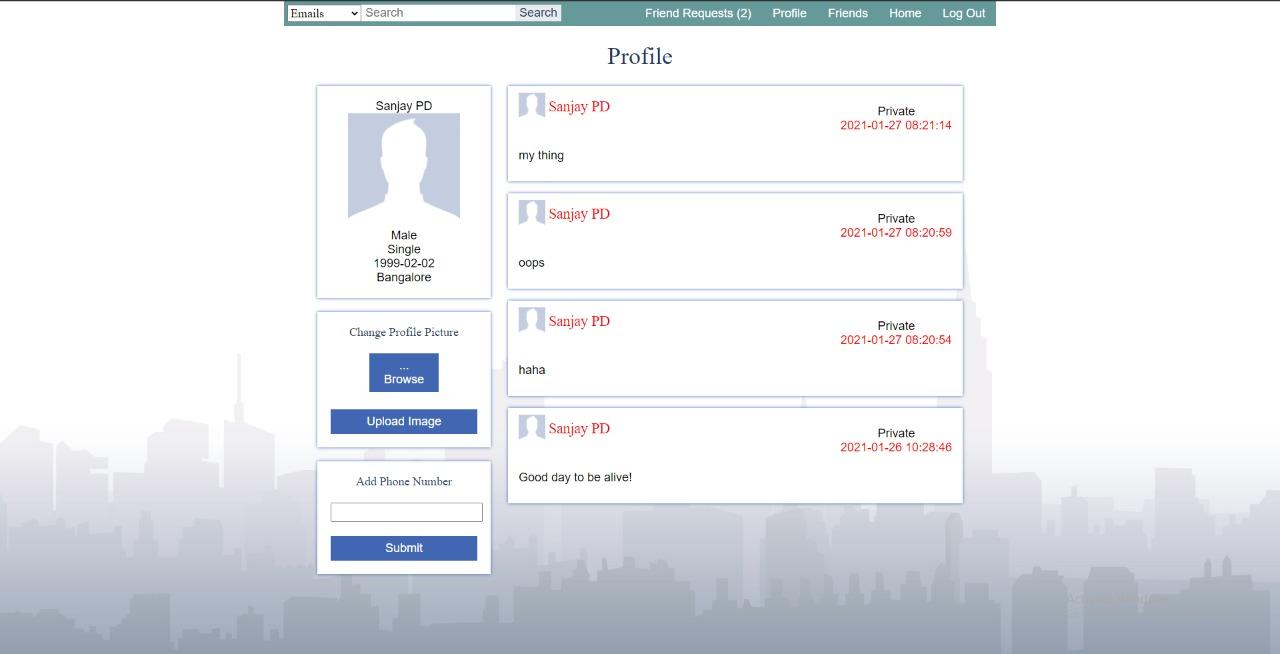
**Figure 6.3 Sociogram Home page** 

#### 

**Figure 6.5 Friend’s Section**

****

**Figure 6.6 Search page**

****

**Figure 6.7 Profile Page**

**NOTE:** The above screenshots are applicable just for only small section from this project. For other sections the screenshots can be generated. But we just took an example of these sections from this Project.

## CHAPTER 7

**TESTING**

* 1. **Testing Techniques**

Software testing techniques help you design better cases. Since exhaustive testing is not possible, testing techniques help reduce the number of test cases to be executed while increasing test coverage. They help identify test conditions that are otherwise difficult to recognize.

## Software Testing Fundamentals

During earlier definition and development of software quality assurance and represents the ultimate review of specification, design and coding. The increasing visibility of software as a system element and attendant “costs” associated with a software failure is motivating force for well planned, through testing.

## Testing Objective

A number of rules that can serve well as testing objectives:

* Testing is a process of executing a program with the intent of finding an error.
* A good test case is one that has high probabilities of finding an undiscovered error. A successful test is one that uncovers as undiscovered error.
* The objective is to design test systematically to uncover different classes of errors and do so with minimum amount of time and effort.

Testing cannot show the absence of defects, it can only show the software defects that are present.

## TEST CASES

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test No.** | **Name of the**  **Page/Section** | **Input** | **Expected**  **Output** | **Obtained**  **Output** | **Remarks** |
| 1 | Search Page | Enter the friend name and Click on Search  button | Direct to  Friend’s  Profile page | Directed to  Friend’s  profile  page | PASS |
| 2 | Friend Requests page | Click on the friend requests  navigation button | Direct to  Friend requests section | Direct to  Friend requests section | PASS |
| 3 | Profile page | Click on the profile button | Shows the  Profile    Page | Shows the Profile page | PASS |
| 4 | Friends Page | Click on the friend’s button | Friends list  should be  displayed | Friends list is displayed. | PASS |
| 5 | Home Page | Click on the home page | All the contents posted should be displayed. | All the posted contents is  displayed. | PASS |
| 6 | Log out page | Click on the Log out page | Return to login page | Returns to login page | PASS |

**CHAPTER 8**

**RESULT AND CONCLUSION**

Sociogram is a website for the social media entertainment and communication where all the contents, friend requests, profile updates could be done. It could also update the profile details which helps the user to update it whenever it is needed. All the login details are stored and thus it could be retained when the login details are given, the passwords should be matched otherwise it would show an error thus maintaining the security of the website. Only the authorized user could update profile details and posts contents after logging-in in to the page.

Thus, all the privacy is maintained accordingly. However, its main usage to provide a social media like interface is obtained and all the security patches are maintained accordingly.

The project has been successfully completed but further it could have been enhanced. I learnt lot of things while doing this project, which can prove very useful in the software field.

As the part of our Academic Project, we have successfully learnt how to create and design a website. And finally, this Project helped us in designing a website, with attractive looking restaurant’s information.

# CHAPTER 9

# BIBILOGRAPHY

## BOOKS:

* + - Randy Connolly, Ricardo Hoar, "Fundamentals of Web Development”, 1st Edition, Pearson Education India PHP: “The Complete Reference”, Steven Holzner, McGraw- Hill Education, 2008.
    - Robin Nixon, “Learning PHP, MySQL &JavaScript with jQuery, CSS and HTML5”, 4th Edition, O’Reilly Publications, 2015.

## WEBSITES:

* + - * [www.tutorialspoint.com](http://www.tutorialspoint.com/)
      * [www.w3schools.com](http://www.w3schools.com/)
      * [www.geeksforgeeks.org](http://www.geeksforgeeks.org/)
      * [www.wikipedia.org](http://www.wikipedia.org/)