



ICLES'

**MOTILAL JHUNJHUNWALA COLLEGE
OF ARTS , SCIENCE & COMMERCE**

Sector 9 A, Vashi, Navi Mumbai – 400 703

**PROJECT REPORT
ON
Social Networking Website**

Submitted to

Mumbai University, Mumbai



UNIVERSITY OF MUMBAI

In the fulfillment of the degree of

Bachelor of Information Technology (B.Sc.CS)

By

Kartik Parmar

**Under the Guidance of
Veena Badgujar**

(2019-2020)



ICLES'

**MOTILAL JHUNJHUNWALA COLLEGE
OF ARTS , SCIENCE & COMMERCE,VASHI**

Certificate

This is to certify that the project
entitled

Social Networking Website

Undertaken by

Kartik.S.Parmar

during the academic year 2019-20. In
fulfillment of B.Sc.CS(Computer Science)
Examination.

It is further certified that he has
completed all required phases of the
project as given by University of Mumbai.

**Internal Guide
Principal**

External Examiner

IT/CS Coordinator

ACKNOWLEDGEMENT

I would especially thank all teaching and non-teaching staff of COMPUTER SCIENCE faculty for inspiring me in completion of project. I am thankful to my project guide **Veena Badgujar** for his timely help and guidance in completing this project successfully.

I would also extend my thanks to our IT/CS Coordinator Mrs. Trupti Kulkarni for her support and facilities provides to us for the same.

Lastly I would like to thank all those who directly and indirectly helped in completion of this project.

Date:

Place:

Completed by,

Kartik Parmar

Table of Content

SR.No	Topic	Page Number
1	Introduction	
1.1	Problem Statement	
1.2	Domain Study	
1.3	Existing Systems	
1.4	Project Scope	
1.5	Organization Profile	
2	SYSTEM REQUIREMENTS	
2.1	Literature Survey	
2.2	Functional Specifications	
2.3	Non-Functional Specification	
2.4	Software Tools Specification	
2.5	Software Specifications	

2.6	Hardware Specifications	
3	Designs	
3.1	Modular Design	
3.2	System Design	
3.2.1	Data Flow Diagram	
3.2.2	Use Case Diagram	
3.2.3	Entity Relationship Diagram	
3.3.4	Database	
3.2.5	Data Directory	
4	Implementation Details	
4.1	Gantt Chart	
4.2	Screen Shots	
5	Testing	
5.1	Test-Cases	
6	Conclusion	
7	References	

1.INTRODUCTION

Social Networking - It's the way the 21st century communicates now. Social networking is the grouping of individuals into specific groups, like small rural communities or a Neighbourhood subdivision. Although social networking is possible in person, especially in the workplace, universities, and high schools, it is most popular online. This is because unlike most high schools, colleges, or workplaces, the internet is filled with millions of individuals who are looking to meet other people.

Social network is the mapping and measuring of relationships and flows between people, groups, organizations, computers, URLs, and other connected information/knowledge entities. The nodes in the network are the people and groups while the links show relationships or flows between the nodes. Social network provides both a visual and a mathematical analysis of human relationships.

Social Networking Website project itself is a huge project comprising various features like profile updating, friend's list organization and various other application to enhance the overall look and feel of the website. However, in this project I am basically working on two essential feature or module (PROFILE MANAGEMENT & FRIENDS ORGANIZATION). PROFILE MANAGEMENT module maintain the profile of a user like name, like, dislikes, hobbies, status etc.

FRIENDS ORGANIZATION module maintains the friend list, handles request and sends request to the other user. Profiles and Friends lists are two key features on social network sites. The third is a public commenting feature ('Testimonials', 'Comments', 'The Wall').

1.1 Problem Statement

We define social network sites as web-based services that allow individuals to construct a public or semi-public profile within a bounded system, articulate a list of other users with whom they share a connection, and view and traverse their list of connections and those made by others within the system. The nature and nomenclature of these connections may vary from site to site.

Since their introduction, social network sites (SNSs) such as MySpace, Facebook, Cyworld and Hi5 have attracted millions of users, many of whom have integrated these sites into their daily practices. As of this writing, there are hundreds of SNSs, with various technological affordances, supporting a wide range of interests and practices. While their key technological features are fairly consistent, the cultures that emerge around SNSs are varied. Most sites support the maintenance of pre-existing social networks, but others help strangers connect based on shared interests, political views, or activities. Some sites cater to diverse audiences, while others attract people based on common language or shared racial, sexual, religious, or nationality-based identities.

Sites also vary in the extent to which they incorporate new information and communication tools, such as mobile connectivity, blogging, and photo/ video-sharing. A lot of business minded people these days are now doing business online and use these social networking sites to respond to customer queries. It isn't just a social media site used to socialize with your friends but also, represents a huge pool of information from day to day living.

A social network service consists of a representation of each user (often a profile), his/her social links, and a variety of additional services. Most social network services

are web-based and provide means for users to interact over the Internet, such as e-mail and instant service, though in a broader sense, social network service usually means an individual centered service whereas online community services are group-centered. Social networking sites allow users to share ideas, activities, events, and interests within their individual networks.

.2 Domain Study

As of May 2013, almost three quarters (72%) of online U.S. adults use social networking sites, up from 67% in late 2012. When we first started asking about social networking sites in February 2005, just 8% of online adults said they used social networking sites. Today, social networking site use is a major activity for internet users from a wide range of demographic groups. Younger adults are especially avid adopters, but social networking continues to grow in popularity for older adults as well. Six out of ten internet users ages 50-64 are social networking site users, as are 43% of those ages 65 and older. Although online seniors are less likely than other age groups to use social networking sites, adoption rates for those 65 and older have tripled in the last four years (from 13% in the spring of 2009 to 43% now). The main types of social networking services are those that contain category places (such as former school year or classmates), means to connect with friends (usually with selfdescription pages), and a recommendation system linked to trust. Popular methods now combine many of Facebook, Google+, YouTube, LinkedIn, Instagram, Pinterest, Tumblr and Twitter widely used worldwide; Nexopia in Canada; Badoo, Bebo, VKontakte (Russia), Delphi (also called Delphi Forums), Draugiem.lv (mostly in Latvia), Hi5 (Europe), Hyves (mostly in The Netherlands), iWiW (mostly in Hungary), Nasza-Klasa, Soup (mostly in Poland), Glocals in Switzerland, Skyrock, The Sphere, StudiVZ (mostly in Germany), Tagged, Tuenti(mostly in Spain), and XING in parts of Europe; Hi5 and Orkut in South America and Central America; Mxit in Africa; and Cyworld, Mixi, Orkut, renren, weibo and Wretch in Asia and the Pacific Islands. Many of these early communities focused on bringing people together to interact with each other through chat rooms, and encouraged users to share personal

information and ideas via personal web pages by providing easy-to-use publishing tools and free or inexpensive web space. Some communities - such as Classmates.com - took a different approach by simply having people link to each other via email addresses. In the late 1990s, user profiles became a central feature of social networking sites, allowing users to compile lists of "friends" and search for other users with similar interests. New social networking methods were developed by the end of the 1990s, and many sites began to develop more advanced features for users to find and manage friends. This newer generation of social networking sites began to flourish with the emergence of SixDegrees.com in 1997 followed by Makeoutclub in 2000, HubCulture and Friendster in 2002 and soon became part of the Internet mainstream. Friendster was followed by MySpace and LinkedIn a year later, and eventually Bebo. Friendster became very popular in the Pacific Island. Orkut became the first social networking in Brazil and than also grow fast in India (Madhavan, 2007). Attesting to the rapid increase in social networking sites' popularity, by 2005, it was reported that MySpace was getting more page views than Google. Facebook, launched in 2004, became the largest social networking site in the world in early 2009. Facebook was first introduced (in 2004) as a Harvard social-networking (Cassidy, 2006).

1.3 Existing Systems

According to the definition above, the first recognizable social network site launched in 1997. SixDegrees.com allowed users to create profiles, list their Friends and, beginning in 1998, surf the Friends lists. Each of these features existed in some form before SixDegrees, of course. Profiles existed on most major dating sites and many community sites. AIM and ICQ buddy lists supported lists of Friends, although those Friends were not visible to others.

Six Degrees was the first to combine these features. SixDegrees promoted itself as a tool to help people connect with and send messages to others. While SixDegrees attracted millions of users, it failed to become a sustainable business and, in 2000, the service closed. Looking back, its founder believes that SixDegrees was simply ahead of its time

(A.Weinreich, personal communication, July 11, 2007). While people were already flocking to the Internet, most did not have extended networks of friends who were online. Early adopters.com explained that there was little to do after accepting Friend requests, and most users were not interested in meeting strangers.

Classmates.com allowed people to affiliate with their high school or college and surf the network for others who were also affiliated, but users could not create profiles or list Friends until years later. From 1997 to 2001, a number of community tools began supporting various combinations of profiles and publicly articulated Friends.

AsianAvenue, BlackPlanet, and MiGente allowed users to create personal, professional, and dating profiles— users could identify Friends on their personal profiles without seeking approval for those connections (O. Wasow, personal communication, August 16, 2007). Likewise, shortly after its launch in 1999,

LiveJournal listed one-directional connections on user pages. LiveJournal's creator suspects that he fashioned these Friends after instant messaging buddy lists (B. Fitzpatrick, personal communication, June 15, 2007)—on LiveJournal, people mark others as Friends to follow their journals and manage privacy settings. The Korean virtual worlds site Cyworld was started in 1999 and added SNS features in 2001, independent of these other sites (see Kim & Yun, this issue). Likewise, when the Swedish web community LunarStorm refashioned itself as an SNS in 2000, it contained Friends lists, guestbooks, and diary pages (D. Skog, personal communication, September 24, 2007).

1.4 Project Scope

- This system provides users to register their various types of profile like social, personal, general, professional.
- This system provides users to send a scrap message, images, and data files to their friends. User can maintain the scrap book whatever scraps he has send to users.
- The system provides user to upload the photos so that user can maintain own Album.
- This system provides user to join the communities according to their scenario.
- This system provides the user to maintain their friend list and user can update their friend list.
- This system provides user to send invitation to another friend and can add to their friend list for future.

1.5 Organization Profile

Incorporated in 2004, as JAVASTREAM Technologies and later got registered as SYSINNOVA InfoTech Pvt. Ltd., it's an ISO 9001:2000 certified organization that operate through well-defined systems and procedures. They have been relentlessly endeavoring to provide end to end solutions to the Information Technology Industry. With our expertise developed through the profound experience we continue contributing in Technology Consulting, Software Development, Training & Talent Management Consulting and Software Development Services to the clients globally. With their headquarters in Bangalore, the IT silicon valley of India, their activities are spread over many locations in Karnataka and other states in India .A formidable manpower of over 58 qualified professionals and strong associations with Global Technology Leaders such as Oracle & SAP are undoubtedly their core strengths. SYSINNOVA InfoTech is an offshore software services and IT consulting company based in Bangalore, India. As a committed outsourcing partner and an IT vendor, our goal is to ensure cost effective, technical excellence and on-time deliveries. While they take care of their end-to-end programming and consulting needs, their clients focus on core business activities which correlate directly to their revenues and profitability. Strategic partnership with them gives their clients the access to latest technology, skilled manpower and scalable team which ultimately results in lower risk and higher ROI. Our core competency lies in web technologies, be it Java-J2EE, Spring, Hibernate, Oracle-XML Publisher, DBA tuning, Oracle Application implementation, PHP and the associated frameworks and CMSs like Joomla, Drupal, SharePoint (MOSS). For over 7 years, they have been working with their clients to bring their creative

ideas onto the web. Today, more than 50 mid sized enterprises and media agencies across the world rely on them to save their invaluable time, resources and money. Their specialized technical and domain expertise can be hired for building modern web sites, web applications and complex enterprise applications.

2. SYSTEM REQUIREMENTS

2.1 Literature Survey

The Web-based social networking services make it possible to connect people who share interests and activities across political, economic, and geographic borders. Through e-mail and instant messaging, online communities are created where a gift economy and reciprocal altruism are encouraged through cooperation. Information is suited to a gift economy, as information is a non rival good and can be gifted at practically no cost.

Facebook and other social networking tools are increasingly the object of scholarly research. Scholars in many fields have begun to investigate the impact of social networking sites, investigating how such sites may play into issues of identity, privacy, social capital, youth culture, and education.

Several websites are beginning to tap into the power of the social networking model for philanthropy. Such models provide a means for connecting otherwise fragmented industries and small organizations without the resources to reach a broader audience with interested users. Social networks are providing a different way for individuals to communicate digitally. These communities of hypertexts allow for the sharing of information and ideas, an old concept placed in a digital environment.

In 2011, HCL Technologies conducted research that showed that 50% of British employers had banned the use of social networking sites/services during office hours.

2.2 Functional Specifications

- **Server Object**

The Server class acts as a wrapper for all server functions for our social networking site. It essentially acts as a link between all of the information such as accounts, account details, pages, notes, etc to our database. When any other model object such as a page is pulled from the server, a temporary copy is made. If that temporary copy is changed in any way. The new version must be sent to the server in order to update the permanent copy. The reason behind local copies is that all the necessary information for the object is sent over in one easy-to-use package. Then the update to the database can be done all at once by sending back that single object. There is no need for multiple functions or a function that takes a large number of parameters.

- **Account Object**

Each user who wants to use the site must create an account. This is the head class that all other objects use to determine what a user does and when the user did it. The account's information has four purposes: hold the login information, hold friend information, hold profile information, and hold privacy information with such a large amount of information to keep track of, the Account class would be very large and difficult to work with. Therefore to ease the load, the Account class was broken up into three different classes. There is the actual account class which keeps track of login information and friend information. It also holds the other two classes within it. Profile information was outsourced to the Account Details class, and privacy settings were outsourced to the Privacy Settings class. The only time the Account class needs to be updated is when the user changes his/her username and/or password.

- **Account Details Object**

An Account Details object is a helper class created whenever a new Account object is created. The object contains all the information that shows up in the user's profile. The user can edit this by modifying his/her profile. Overall, this class has no other purpose but to be a helper class to its account object.

- **Privacy Settings Object**

A Privacy Settings object is the other helper class created whenever a new Account object is created. This object contains all the privacy settings that a user has, such as who can view his/her media or custom pages. This class is called any time a user visits a profile or content created by another user. However, it does not directly interact with the other model classes, only the view.

- **Chat Session Object, Event Object, Link Object, Note Object and Page Object::**

These objects contain unique information for a particular type of action a user performs. All of these contain a reference to the account that owns them. Each object is a ?workingcopy? of an object in the Server. Anytime one of these objects is created on the Server, an entry of its creation is added to the news feed database.

- **Wall Post Object**

A Wall Post object works very similar to a Message object. The only difference is that the Wall Post objects are viewable to all friends of the user. Another difference is that a Wall Post object can hold comments. If a user adds a wall post to his or her own account, their status will change to the new wall post.

- **Followers**

A follow represents a user who chooses to see all of another user's posts in their content feed. Getting users to follow their accounts is a primary objective for online businesses with a social media presence.

Twitter, Instagram and Facebook all have a form of news feed that delivers content to their users. The content of news feeds vary by platform, but they consist mainly of the posts, articles and images from other people whom the user has chosen to follow.

Follows are distinct from likes and shares. Whereas likes, retweets and shares indicate that people find value in an individual post, a follow expresses interest in receiving constant updates. A follow indicates a higher level of engagement with the audience, giving it more value than

- **Account Creation**

When a user accesses the site for the first time, he/she must create an account before using any of the site features. The account creation process is broken into three sections. The first section deals with the login information and is required for the user to fill out.

This includes the email, password, and password confirmation. The purpose behind the password confirmation is to ensure that the user didn't accidentally mistyped when creating a password. The second section deals with information about who you are, such as name, location, and gender. Most of these fields are optional except for your name and gender. It wouldn't be much of a social network if everyone was named anonymous. The final section deals with information about the users likes and

dislikes,

such as interests and activities. Unlike the other two sections, this section is completely optional. Once the user clicks create account, a new account, account details, and privacy settings are added to the server, and the user is brought back to the login page

- **Messaging**

Unlike with chatting, users can send a message to any other user. For user A to send a message to user B, he/she simply goes to the message center tab and clicks compose message. He then fills out the form as shown in the figure below with the email of the recipient, the message title, and the message content. The new message will then be in User B's inbox in the message center.

Messaging is implemented by storing a record of the message in the server. First a message object is constructed that takes in the to and from accounts, the message body, and the date it was sent. The message body is retrieved from a user submitted form, the from account is retrieved from the user session, and the receiver account is retrieved from the recipient email address. The receiver account is validated, to ensure the sender didn't try to send a message to a non-existent receiver. Finally, the new message is stored in the server. Retrieving messages sent to the user is a simple method of querying the server for any messages that have been sent to the user, and this is done by checking the receiver id of each message against the account id of the user. Any matches are returned to the inbox of the user as shown in the figure below.

- **Media Uploading**

As people use their social networking account, they will want to be able to upload funny or interesting images, video, music, etc., to share with their friends. The media upload section will be located at the media tab, where users are able to specify a file to upload as well as provide a short description of the file to be uploaded. In order to prevent users from uploading potentially malicious files such as executables, only certain file extensions are supported. These allowed extensions cover popular image extensions such as png, jpg, gif, and bitmap, video extensions wmv and avi, audio extensions mp3, wma, and wav, and some document files including txt, rtf, doc, and pdf. The inner workings of this uploading process are surprisingly simple, since the HTTPserver takes care of requesting the file from the client automatically. After the user clicks upload, the file is sent to the server and stored in a temporary location. Through PHP all information about these temporary files can be accessed through the \$ FILES variable. The temporary location is stored in that variable and can be used to pull the file name and file extension.[1] That extension is then checked against an array of allowed extensions.

If the extension is in the list, the file is then moved to a permanent location in the media folder under a subfolder for the user's account. If the extension is not on the list, it is left in the temporary folder where it will be automatically deleted by the server once the php script completes.

- **Other features**

All the other features, such as the wall pages, and notes are fairly similar. Their main purpose is to add additional places to have places for additional content. All are created using similar methods, and are stored similarly but are used for different purposes.

1. Wall

The wall is a place where the account owner can express his current feeling or thoughts with a wall status. Other users can add their thoughts by creating wall posts that show up below the status. It is basically a way for many users to communicate publicly or simply leave their thoughts about each other.

2. Notes

Notes are very similar to wall posts with one major exception. Unlike wall posts which can be created by anyone, only the account owner can create notes. It's main purpose is for the account owner to create reminders for him/herself, or create announcements for any other users to see.

2.3 Non-Functional Specification

- Secure access of confidential data by user name and password. This application is secure for every kind of its users, because if any user logout from any session then nobody will be able to access his profile without knowing his confidential password.

- 24 X 7 availability

- Better component design to get better performance at peak time.

- The database used here is robust, reliable & fast. So users will have to wait for the output very short time.

- This application can be accessed from any type of platform.

- There is no case of redundancy in the database so it will not take extra memory space.

- Username & password are sent to the users via email after registration.

- Password recovery system is also provided in case of forgetting the password.

2.4 Software Tools Specification

MySQL:

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a fullfeatured database management system often use MySQL.

MySQL can be built and installed manually from source code, but this can be tedious so it is more commonly installed from a binary package unless special customizations are required. On most Linux distributions the package management system can download and install MySQL with minimal effort, though further configuration is often required to adjust security and optimization settings.

Though MySQL began as a low-end alternative to more powerful proprietary databases, it has gradually evolved to support higher-scale needs as well. It is still most commonly used in small to medium scale single-server deployments, either as a component in a LAMP-based web application or as a standalone database server. Much of MySQL's appeal originates in its relative simplicity and ease of use, which is enabled by an ecosystem of open source tools such as phpMyAdmin. In the medium range, MySQL can be scaled by deploying it on more powerful hardware, such as a multi-processor server with gigabytes of memory.

There are however limits to how far performance can scale on a single server ('scaling up'), so on larger scales, multi-server MySQL ('scaling out') deployments are required to provide improved performance and reliability. A typical high-end configuration can be replicated to multiple slaves that handle all read operations. The

master server synchronizes continually with its slaves so in the event of failure a slave can be include a powerful master database which handles data write operations and promoted to become the new master, minimizing downtime. Further improvements in performance can be achieved by caching the results from database queries in memory using memcached, or breaking down a database into smaller chunks called shards which can be spread across a number of distributed server clusters.

PHP Language

PHP is a scripting language designed to fill the gap between SSI (Server Side Includes) and Perl, intended for the web environment. Its principal application is the implementation of web pages having dynamic content. PHP has gained quite a following in recent times, and it is one of the frontrunners in the Open Source software movement. Its popularity derives from its C-like syntax, and its simplicity. PHP is currently divided into two major versions: PHP 4 and PHP 5, although PHP 4 is deprecated and is no longer developed or supplied with critical bug fixes. PHP 6 is currently under development. PHP was designed by Rasmus Lerdorf to display his resume online and to collect data from his visitors.

PHP allows a static webpage to become dynamic. "PHP" is an acronym that stands for "PHP: Hypertext Preprocessor". The word "Preprocessor" means that PHP makes changes before the HTML page is created. This enables developers to create powerful applications which can publish a blog, remotely control hardware, or run a powerful website such as Wikipedia or Wikibooks. Of course, to accomplish something such as this, you need a database application such as MySQL.

PHP code is interpreted by a web server with a PHP processor module, which generates the resulting web page: PHP commands can be embedded directly into an HTML source document rather than calling an external file to process data. It has also evolved to include a interface capability and can be used in standalone graphical applications.

The PHP language was originally implemented as an interpreter, and this is still the most popular implementation. Several compilers have been developed which decouple the PHP language from the interpreter. Advantages of compilation include better execution speed, static analysis, and improved interoperability with code written in other languages. PHP includes free and open source libraries with the core build. PHP is a fundamentally Internet-aware system with modules built in for accessing File Transfer Protocol (FTP) servers, many database servers, embedded SQL libraries such as embedded PostgreSQL, MySQL, Microsoft SQL Server and SQLite, LDAP servers, and others. Many functions familiar to C programmers such as those in the studio family are available in the standard PHP build.

JavaScript:

(JS) is a lightweight, interpreted, or just-in-time compiled programming language with first-class functions. While it is most well-known as the scripting language for Web pages, many non-browser environments also use it, such as Node.js, Apache CouchDB and Adobe Acrobat. JavaScript is a prototype-based, multi-paradigm, dynamic language, supporting object-oriented, imperative, and declarative (e.g. functional programming) styles. [Read more about JavaScript.](#)

This section is dedicated to the JavaScript language itself, and not the parts that are specific to Web pages or other host environments. For information about APIs specific to Web pages, please see [Web APIs](#) and [DOM](#). The standard for JavaScript is ECMAScript. As of 2012, all modern browsers fully support ECMAScript 5.1. Older browsers support at least ECMAScript 3. On June 17, 2015, ECMA International published the sixth major version of ECMAScript, which is officially called ECMAScript 2015, and was initially referred to as ECMAScript 6 or ES6. Since then, ECMAScript standards are on yearly release cycles. This documentation refers to the latest draft version, which is currently ECMAScript 2020.

Back End: PHP MyAdmin

PHPMyAdmin is a free software tool written in PHP, intended to handle the administration of MySQL over the Web. PhpMyAdmin supports a wide range of operations on MySQL, Maria DB and Drizzle. Frequently used operations (managing databases, tables, columns, relations, indexes, users, permissions, etc.) can be performed via the user interface, while you still have the ability to directly execute any SQL statement.

Features provided by the program include:

1. Web interface
2. MySQL database management
3. Import data from CSV and SQL
4. Export data to various formats: CSV, SQL, XML, PDF (via the TCPDF library), ISO/IEC 26300 - Open Document Text and Spreadsheet, Word, Excel, LaTeX and others
5. Administering multiple servers
6. Creating PDF graphics of the database layout
7. Creating complex queries using Query-by-Example (QBE)
8. Searching globally in a database or a subset of it
9. Transforming stored data into any format using a set of predefined functions, like displaying BLOB-data as image or download-link
10. Live charts to monitor MySQL server activity like connections, processes,

2.5 Hardware Requirements

Processor : Intel Pentium IV 2.0 GHz and above.

RAM : 512 MB and above.

Hard disk : 80GB and above.

Monitor : CRT or LCD monitor.

Keyboard : Normal or Multimedia.

Mouse : Compatible mouse.

2.6 Software Requirements

Front End : XAMPP

Language : PHP & JAVASCRIPT.

Back End : PHP MyAdmin.

Operation System : Windows XP or above.

Browser : Any latest browser.

3.1 Modular Design

The application comprises the following major modules:

Register to be a member Module.

This module provides functionalities for those people who want to open an account. Applicants can post their views with personal and professional details. They can also update the profile as frequently as required. The member can also browse through the profile's available. Members can also get message alerts when someone messages them.

Profile Module.

This module provides functionalities related to members' profile. Logged users can see their details and if they wish to change any of their information they can edit it.

3.2 System Design

3.2.1 Data Flow Diagrams

Data flow diagrams model the flow of data into, through, and out of an information system:

- show the processes that change or transform data
- show the movement of data between processes
- represent a system as a network of processes which transform data flowing between

The user screen flow shows what a user of the community will see. After successfully logging on, the user will be given various links (such as search users, search boards, view mail, etc.), and be able to select options from there, or go back to their home.

Use case diagrams.

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems.

System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering,

catalog updating, payment processing, and customer relations. A use case diagram contains four components.

The boundary, which defines the system of interest in relation to the world around it.

- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which are the specific roles played by the actors within and around the system.
- The relationships between and among the actors and the use cases

- **Login/Registration**



Fig 3.2 Login/Registration

- **Profile Page**

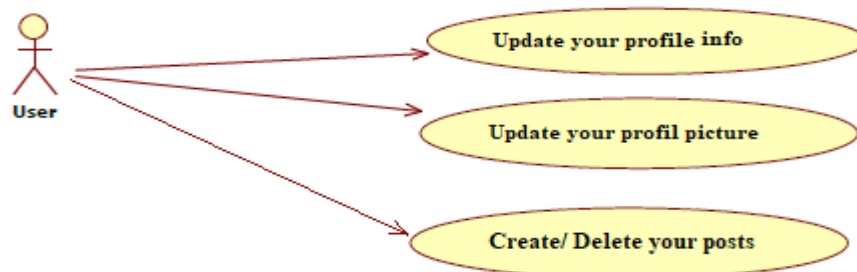


Fig 3.4

- Search Page

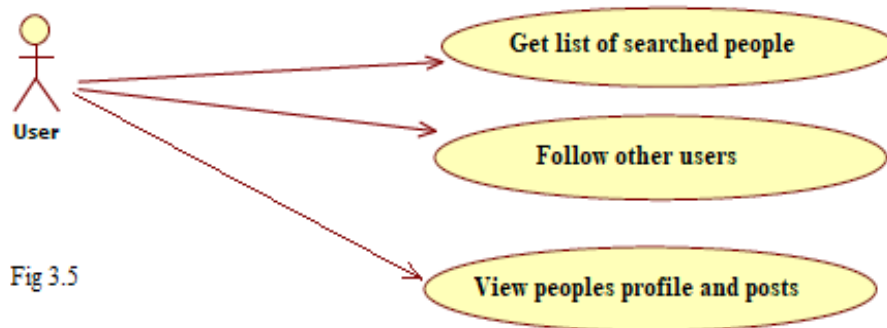


Fig 3.5

- Home Page

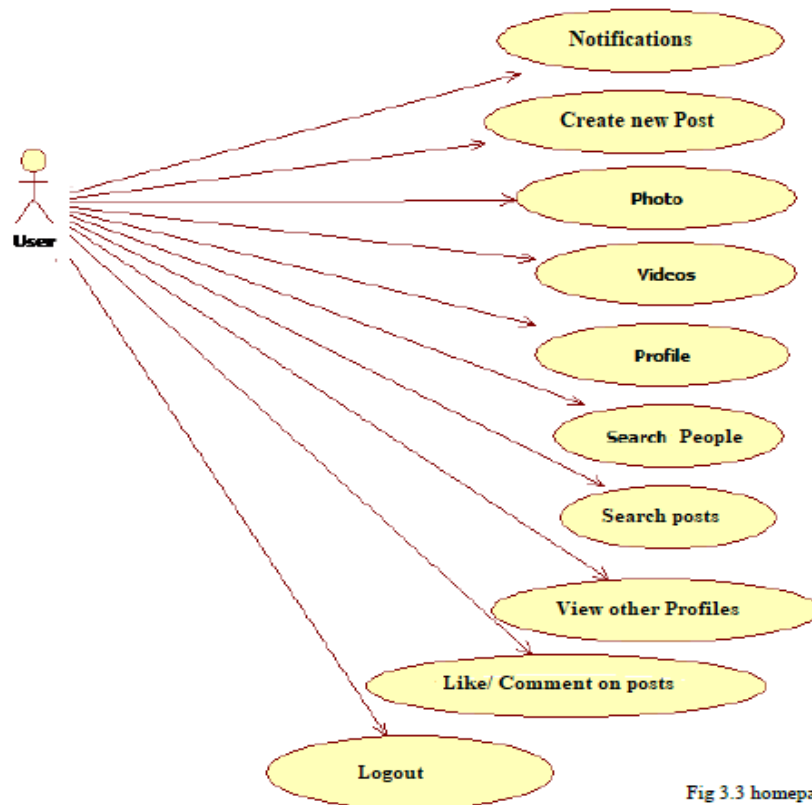


Fig 3.3 homepage

- Others Profile page

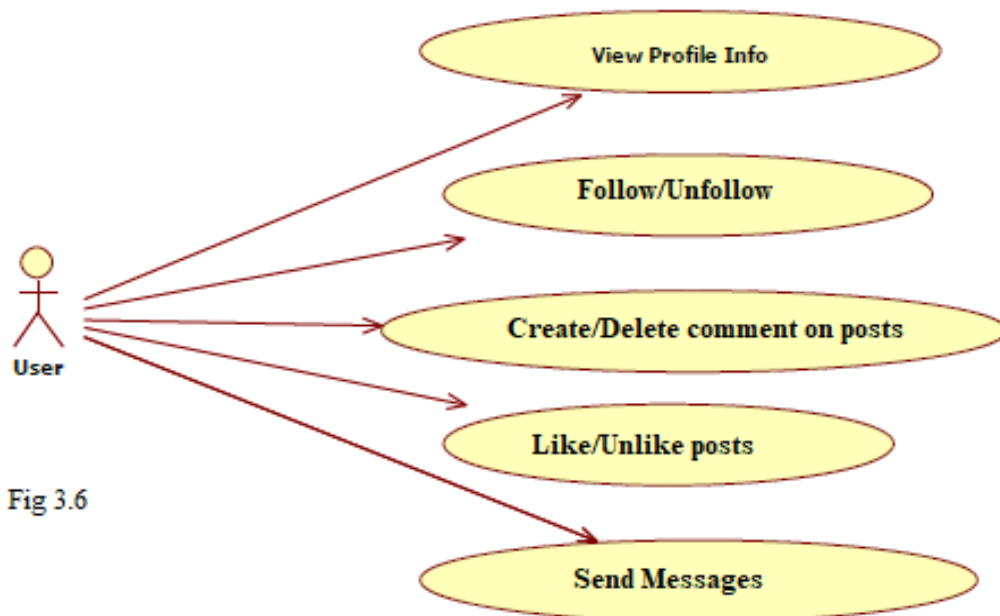


Fig 3.6

- Create-Post Page

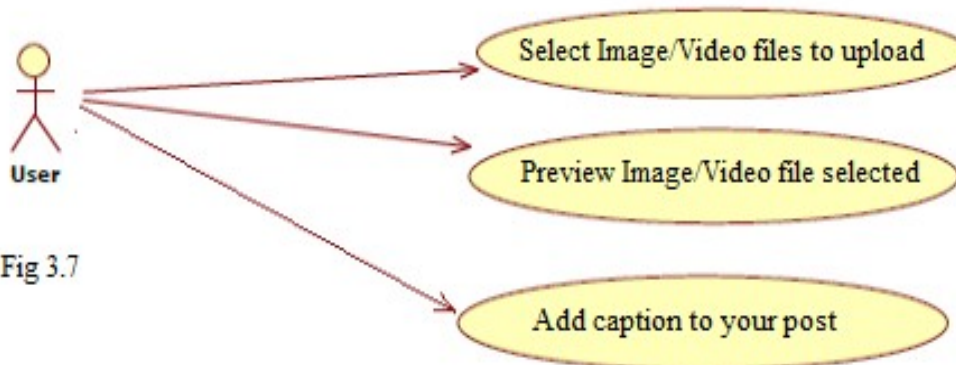


Fig 3.7

3.2.2 Context Diagram

The highest level data flow diagram is the context diagram.

- The context diagram shows the interaction of the system with its environment in terms of data flows
- The context diagram defines the boundary of the system (the scope of the system)
- Only the data flows which leave the system and the data flows which come from outside the system are shown.

0 Level DFD: A level 0 DFD, also called a fundamental system model or context diagram represents the entire software element as a single bubble with input and output data indicated by incoming and outgoing arrows, respectively.

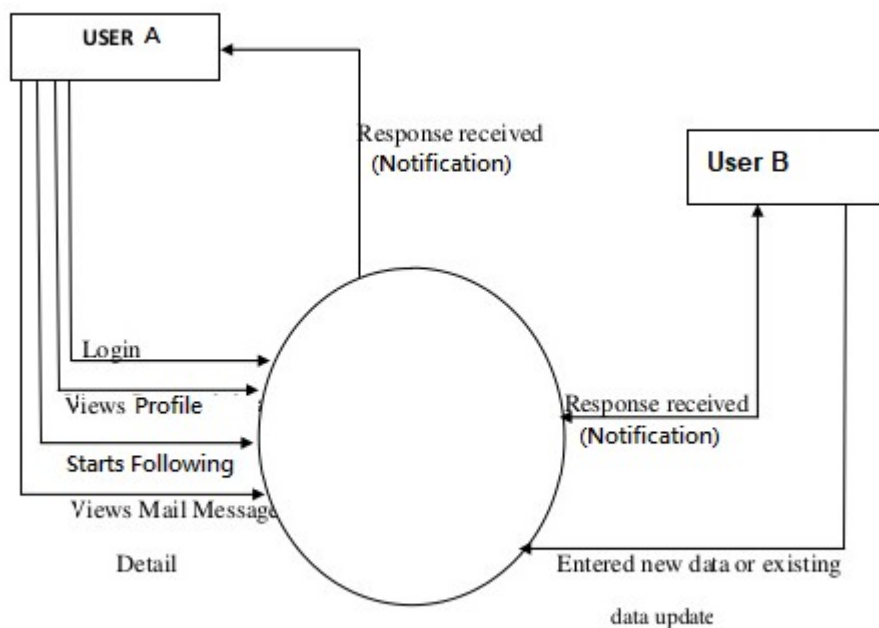
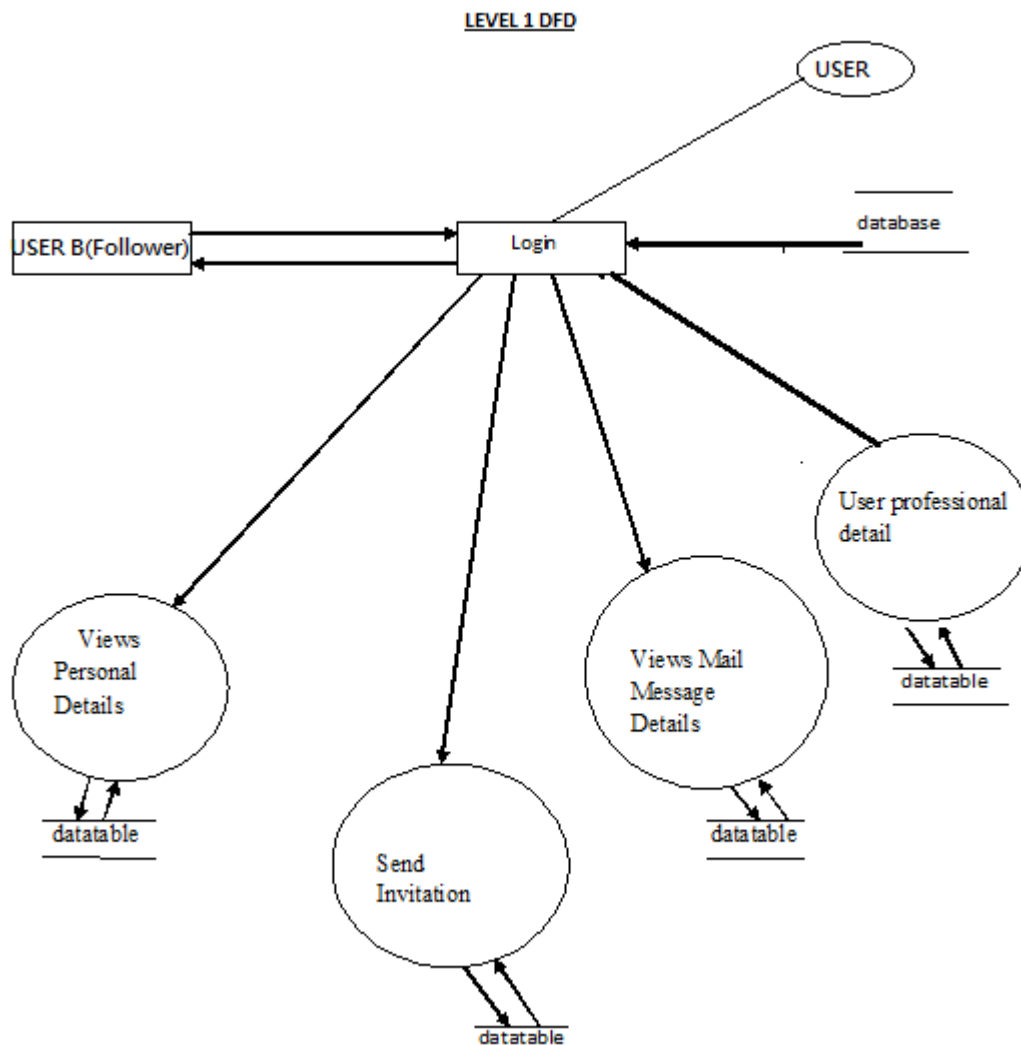


Fig 3.10 0 level DFD

1 Level DFD:

This level of DFD provide more detailed structure. It provides a detailed view of requirements and flow of data from 1 bubble to another.



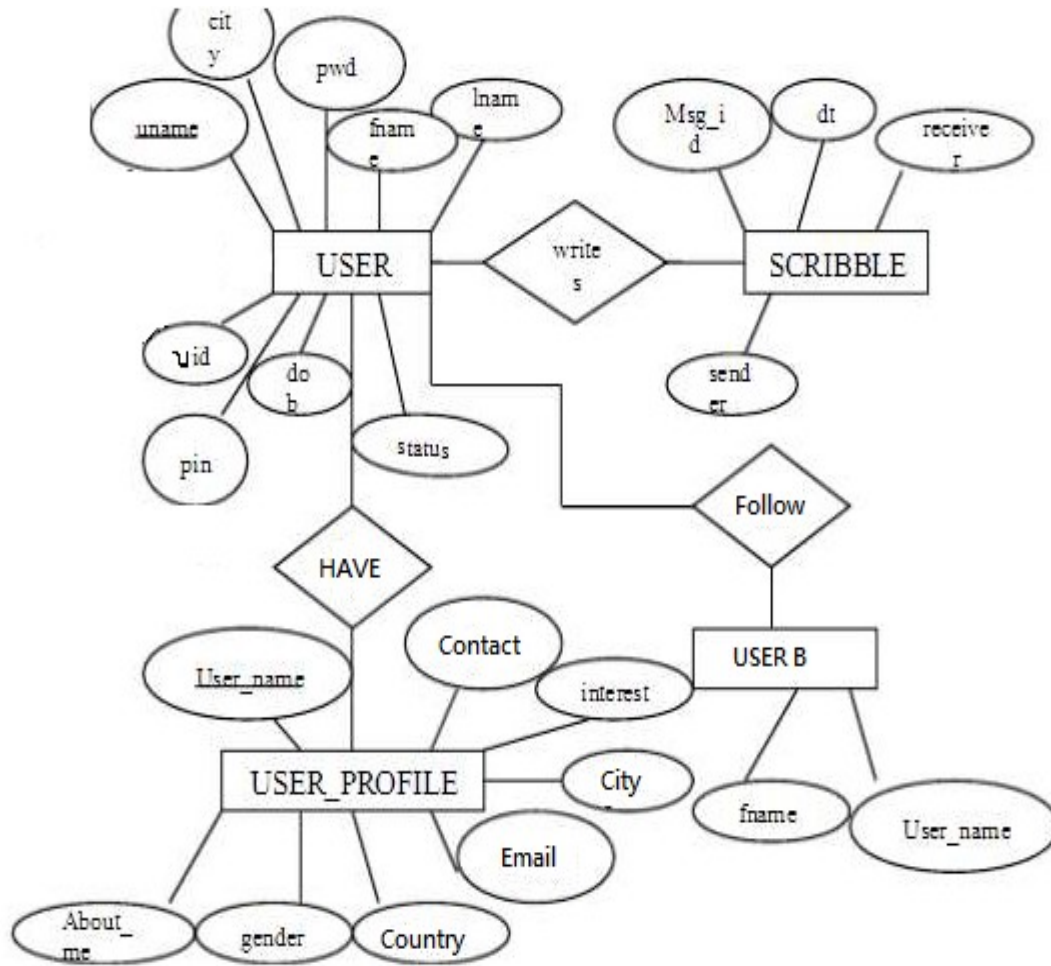
3.2.3 ENTITY RELATIONSHIP DIAGRAM

The entity relationship model is a high level data model. It is based on a perception of a real world that consists of a collection of basic objects, called entities, and of relationship among these objects. It was developed to facilitate database design by allowing specification of an enterprise schema, which represent the overall logical structure of a database.

Entity: An entity is an object that has its existence in the real world. It includes all those -things? about which data is collected. An entity may be a tangible object such as a student, a place or a part. It may also be non-tangible such as an event, a job title or a customer account. For example, if we say that a customer buys goods, it means customer and goods are entities. Diagrammatically, entities are represented in rectangles. An Entity Set: It is a set of entities of the same type that share the same properties, or attributes. The set of all persons who are customers at a given bank, example, can be defined as the entity set customer.

Attributes: Attributes are units that describe the characteristics or properties of entities. In a database, entities are represented by tables and attributes by columns. The goods entity may have attributes like code and price. They are drawn in elliptical shapes along with the entity rectangle. The entity relationship diagram of mailing system is drawn on the next page:

ENTITY RELATIONSHIP DIAGRAM

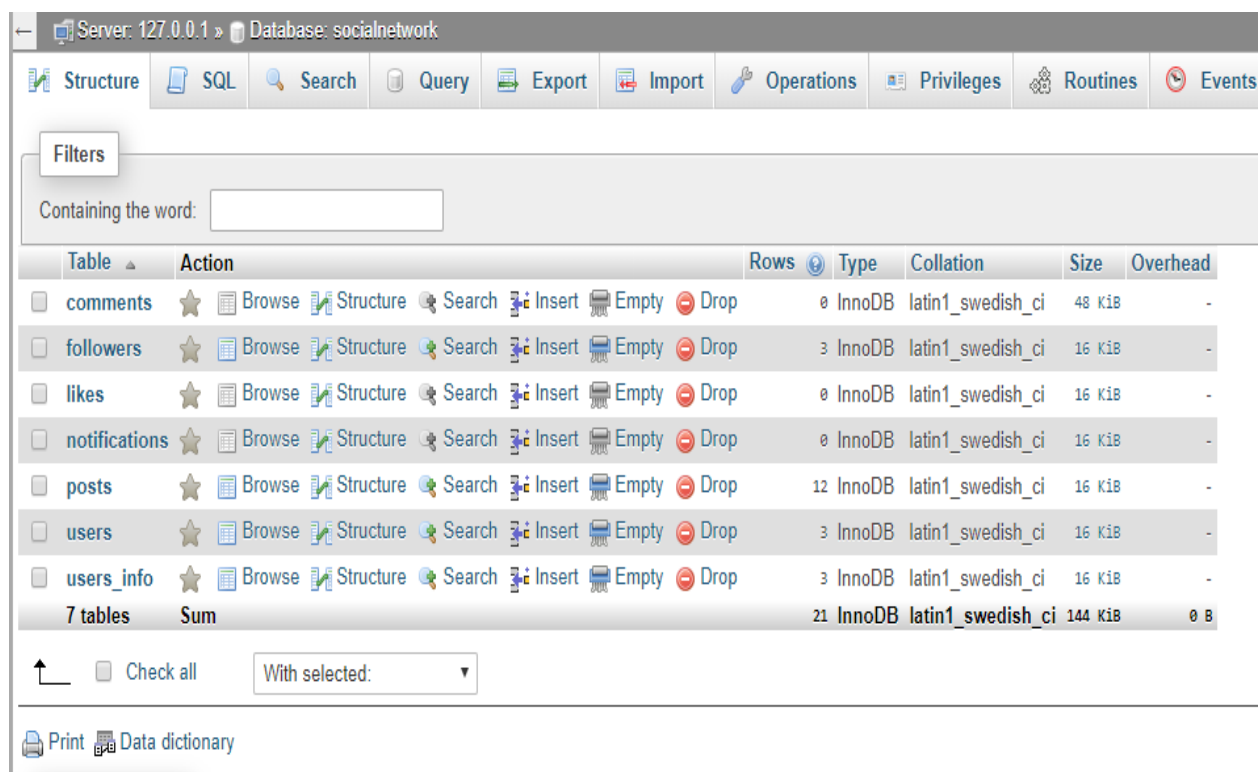


3.2.4 Database

Data base is used to store the relevant information of the individuals. A database is a collection of rows and columns in which rows indicates the tuple and column indicates the domain of table. Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters. Need to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity. The term database design can be used to describe many different parts of the design

of an overall database system.

Principally, and most correctly, it can be thought of the logical design of the relation of the base data structures used to store the data. In the relational model these are the classes and named relationships. However, the term database design could also be used to apply to overall process of designing, not just the base data structure, but also the forms and queries used as part of the overall database application within the database management system (DBMS).



The screenshot shows a database management interface for a server at 127.0.0.1, connected to a database named 'socialnetwork'. The interface includes a top menu bar with options: Structure, SQL, Search, Query, Export, Import, Operations, Privileges, Routines, and Events. Below the menu is a 'Filters' section with a search box labeled 'Containing the word:'. The main area displays a table list with columns: Table, Action, Rows, Type, Collation, Size, and Overhead. The table list contains seven tables: comments, followers, likes, notifications, posts, users, and users_info. Each table row includes a checkbox, a star icon, and a set of action icons (Browse, Structure, Search, Insert, Empty, Drop). The 'posts' table has 12 rows, while the others have 3 or 0 rows. The 'Sum' row at the bottom indicates there are 7 tables in total, with a combined size of 144 KiB and 0 B overhead. At the bottom of the interface, there is a 'Check all' checkbox and a 'With selected:' dropdown menu, along with 'Print' and 'Data dictionary' links.

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> comments	★ Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	48 KiB	-
<input type="checkbox"/> followers	★ Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 KiB	-
<input type="checkbox"/> likes	★ Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 KiB	-
<input type="checkbox"/> notifications	★ Browse Structure Search Insert Empty Drop	0	InnoDB	latin1_swedish_ci	16 KiB	-
<input type="checkbox"/> posts	★ Browse Structure Search Insert Empty Drop	12	InnoDB	latin1_swedish_ci	16 KiB	-
<input type="checkbox"/> users	★ Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 KiB	-
<input type="checkbox"/> users_info	★ Browse Structure Search Insert Empty Drop	3	InnoDB	latin1_swedish_ci	16 KiB	-
7 tables	Sum	21	InnoDB	latin1_swedish_ci	144 KiB	0 B

3.2.5 Data Dictionary

Table 3.1 Comment

FIELD	TYPE	NULL	Key	Default	Extra
id	int(11) unsigned	NO	PRI	NULL	auto_increment
comment	text	NO		NULL	
user_id	int(11) unsigned	NO	MUL	NULL	
posted_at	datetime	NO		NULL	
post_id	int(11) unsigned	NO	MUL	NULL	

Table 3.2 likes

FIELD	TYPE	NULL	Key	Default	Extra
id	int(11) unsigned	NO	PRI	NULL	auto_increment
pid	int(11) unsigned	NO	MUL	NULL	
user_name	varchar(32)	NO		NULL	

Table 3.3 Users

FIELD	TYPE	NULL	Key	Default	Extra
id	int(11) unsigned	NO	PRI	NULL	auto_increment
username	varchar(32)	NO		NULL	
password	varchar(60)	NO		NULL	
email	text	NO		NULL	
profilepic	varchar(60)	YES		NULL	

Table 3.4 posts

FIELD	TYPE	NULL	Key	Default	Extra
id	int(11) unsigned	NO	PRI	NULL	auto_increment
body	varchar(32)	YES		NULL	
posted_at	datetime	NO		NULL	
user_name	varchar(32)	NO		NULL	
image	varchar(60)	NO		NULL	
likes	int(11)	YES		NULL	

Table 3.5 followers

FIELD	TYPE	NULL	Key	Default	Extra
id	int(11) unsigned	NO	PRI	NULL	auto_increment
follower	varchar(32)	NO		NULL	
following	varchar(32)	NO		NULL	

Table 3.6 notifications

FIELD	TYPE	NULL	Key	Default	Extra
id	int(11) unsigned	NO	PRI	NULL	auto_increment
message	text	NO		NULL	
sender	varchar(32)	NO		NULL	
receiver	varchar(32)	NO		NULL	

Table 3.7 notifications

FIELD	TYPE	NULL	Key	Default	Extra
id	int(11) unsigned	NO	PRI	NULL	auto_increment
username	varchar(32)	NO		NULL	
email	text	YES		NULL	
country	varchar(32)	YES		NULL	
city	varchar(32)	YES		NULL	
dob	date	NO		NULL	
phone	varchar(14)	YES		NULL	
bio	varchar(30)	YES		NULL	

4 IMPLEMENTATION

index.php(Homepage)

```
<?php
session_start();
//PDO
connection_____

_____

try{
    $connect = new PDO('mysql:host=127.0.0.1; dbname=socialnetwork;','root','kartik');
        $connect->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
        $username=$_SESSION["username"];
}

    catch (PDOException $e) {
echo 'Connection failed: ' . $e->getMessage();
}

//posting
code_____

_____

    if(isset($_POST['post'])){
        if(empty($_POST['postbody']))
        {
            $message = '<label>Cant post nothing XD</label>';
```

```
}
```

```
else
```

```
{ $postbody=$_POST['postbody'];
```

```
    $query = "INSERT INTO posts (body,posted_at,user_name) VALUES  
('$postbody',NOW(),'$username')";
```

```
    $connect->exec($query);
```

```
}
```

```
}
```

```
// post retrieval
```

```
code_____
```

```
_____
```

```
    $sql = "SELECT * FROM posts ORDER BY id DESC";
```

```
    $q = $connect->prepare($sql);
```

```
    $q->execute( array('user_name' => $_SESSION["username"]));
```

```
    $q->setFetchMode(PDO::FETCH_ASSOC);
```

```
// on click search
```

```
code_____
```

```
_____
```

```
if(isset($_POST['search']))
```

```
{
```

```
    $var1 = $_POST['var1'];
```

```
    $_SESSION['var1'] = $var1;
```

```
    $query = "SELECT * FROM users WHERE username LIKE :search OR
```

```

email LIKE :search";
$stmt = $connect->prepare($query);
$stmt->bindValue(':search', '%' . $var1 . '%', PDO::PARAM_INT);

$stmt->setFetchMode(PDO::FETCH_ASSOC);

        header('Location: search.php');
    }

    if(isset($_POST['comment']))
    {
        $sender=$_SESSION["username"];
        $comment= $_POST['comment'];
        $query = "INSERT INTO comment (comment,sender,receiver) VALUES
('$comment',$sender,'$reciever')";
        $connect->exec($query);
    }
//PHP
end_____

```

```

_____  

?>  

<html>  

<head>  

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

<style>  

<?php include 'log.css'; ?>

```

```
</style>
</head>
<body>
<div class="header">
<a href="#default" class="logo">Welcome <?php echo $_SESSION["username"]; ?
> </a>

<div class="header-right">
<a href="logout.php">Logout</a>
<a href="pf.php">Profile</a>
<a href="#about">Settings</a>
</div>
<div style="padding-left:400px;">
<form name="frmSearch" method="post" action="index.php">
<input name="var1" type="text" id="var1">
<input type="submit" value="Search" name="search">
</div>
</form>
</div>
<div>
<div style="padding-bottom:20px; margin-bottom:2px;">
<br>
<a href="pic.php" style="font-family:Segoe Script; font-size:25px; text-
decoration:none; color:20F0EA; padding-left:90px;">Create a new Post, Share your
content with the world..... Click here</a>
```

```

<hr style="border:solid 1px gray">
</div>
<div id="container" style="padding-left:20px;margin-top:40px">
<h1>News Feed</h1>
<div id="columns">
<?php while ($row = $q->fetch()): ?>
<figure>

    <?php echo htmlspecialchars($row['user_name']); ?> </b>
    
<figcaption><?php echo htmlspecialchars($row['body']); ?> </br>Posted at: <?php
echo htmlspecialchars($row['posted_at']); ?> <hr> </figcaption>

    <hr>

    <script>
$( "#search" ).click(function(e) {
    e.preventDefault();
    $(".search_box").toggleClass('active');
});
</script>

<input type="submit" class="" value="S"> &nbsp; &nbsp;
<input type="text" name="comment" placeholder="Comment....."> <input
type="submit" class="" value="Post">

</figure>

<?php endwhile; ?>

```


<small>Art © Claire Hummel </small>

</div> </div>

</body>

</html>

Profile.php(profile page)

```
<?php
```

```
    session_start();
```

```
    $message = "";
```

```
    // PDO
```

```
connection_____
```

```
    $connect = new PDO('mysql:host=127.0.0.1;
```

```
dbname=socialnetwork;','root','kartik');
```

```
    $connect->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
```

```
    $username=$_SESSION["username"];
```

```
    // Button click Post
```

```
code_____
```

```
    if(isset($_POST['ok'])){\
```

```
//image code_____
```

```
        $folder ="uploads/";
```

```
        $image = $_FILES['image']['name'];
```

```
        $path = $folder . $image ;
```

```
        $target_file=$folder.basename($_FILES["image"]["name"]);
```

```
        $imageFileType=pathinfo($target_file,PATHINFO_EXTENSION);
```

```
        $allowed=array('jpeg','png','jpg'); $filename=$_FILES['image']['name'];
```

```
        $ext=pathinfo($filename, PATHINFO_EXTENSION); if(!in_array($ext,$allowed) )
```

```
        {\
```

```
echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";
```

```
}
```

```
Else
```

```
{
```

```
//caption code_____
```

```
$postbody=$_POST['postbody'];
```

```
move_uploaded_file( $_FILES['image'] ['tmp_name'], $path);
```

```
$query = "INSERT INTO posts (body,posted_at,user_name,image)
```

```
VALUES ('$postbody',NOW(),'$username','$image')";
```

```
$connect->exec($query);
```

```
}
```

```
}
```

```
//Profile card
```

```
code_____
```

```
$query = "SELECT * FROM users WHERE username = :user_name";
```

```
$z = $connect->prepare($query);
```

```
$z->execute( array('user_name' => $_SESSION["username"]));
```

```
$z->setFetchMode(PDO::FETCH_ASSOC);
```

```
//Post retrieval
```

```
code_____
```

```
$query = "SELECT * FROM posts WHERE user_name = :user_name ORDER BY  
id DESC";
```

```
$q = $connect->prepare($query);
```

```
    $q->execute( array('user_name' => $_SESSION["username"]));
```

```
$q->setFetchMode(PDO::FETCH_ASSOC);
```

```
    //Profile Card info retrieval
```

```
code_____
```

```
$e="";
```

```
    try{
```

```
        $query = "SELECT * FROM users_info WHERE username = :user_name ORDER  
BY id DESC";
```

```
        $x = $connect->prepare($query);
```

```
        $x->execute( array('user_name' => $_SESSION["username"]));
```

```
$x->setFetchMode(PDO::FETCH_ASSOC);
```

```
    }
```

```
    catch (Exception $e) {
```

```
        // Nothing, this is normal
```

```
    }
```

```
    $pc = $connect->query("select count(*) from posts ")->fetchColumn();
```

```
?>
```

```
<head>
```

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

```
<style>
```

```
<?php include 'profilecard.css'; ?>
```

```
</style>
```

```
<link href="//maxcdn.bootstrapcdn.com/bootstrap/3.3.0/css/bootstrap.min.css"
rel="stylesheet" id="bootstrap-css">
<script src="//maxcdn.bootstrapcdn.com/bootstrap/3.3.0/js/bootstrap.min.js"> </
script>
<script src="//code.jquery.com/jquery-1.11.1.min.js"> </script>
<!-- Include the above in your HEAD tag -->
```

```
</head>
```

```
<!-- JAVA-SCRIPT
```

```
CODE----->
```

```
<script>
```

```
$(document).ready(function() {
    var $btnSets = $('#responsive'),
    $btnLinks = $btnSets.find('a');
    $btnLinks.click(function(e) {
        e.preventDefault();
        $(this).siblings('a.active').removeClass("active");
        $(this).addClass("active");
        var index = $(this).index();
        $("div.user-menu>div.user-menu-content").removeClass("active");
        $("div.user-menu>div.user-menu-content").eq(index).addClass("active");
    });
});
```

```
$( document ).ready(function() {
```

```

$("[rel='tooltip']").tooltip();

$('.view').hover(
    function(){
        $(this).find('.caption').slideDown(250); //.fadeIn(250)
    },
    function(){
        $(this).find('.caption').slideUp(250); //.fadeOut(205)
    }
);
});

```

```

$( "#search" ).click(function(e) {
    e.preventDefault();
    $(".search_box").toggleClass('active');
});

```

</script>

<!-- Header code

----->

<div class="header">

<h1> Project </h1>

<div> <nav>

 ☌

<form class="search_box" id="search_box" action="/search/">

```
<input name="search_criteria" placeholder="Search by keyword" value=""
type="text">
```

```
<input class="search_icon" value="Search" type="submit">
```

```
</form>
```

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

```
</div>
```

```
<div style="padding-bottom:20px; margin-bottom:2px;">
```

```
</br>
```

```
<a href="pic.php"style="font-family:Segoe Script; font-size:25px; text-
decoration:none; color:20F0EA; padding-left:90px;">Create a new Post, Share your
content with the world...Click here.</a>
```

```
<hr style="border:solid 1px gray">
```

```
</div>
```

```
<!-- Profile card
```

```
-----
----->
```

```
<div class="container">
```

```
<div class="row user-menu-container square">
```

```
<div class="col-md-7 user-details">
```

```
<div class="row coralbg white">
```

```
<div class="col-md-6 no-pad">
```

```
<div class="user-pad">
```

```
<h3>Welcome back,<?php echo $_SESSION["username"];?></h3>
```

```

        <?php while ($row = $x->fetch()): ?>
<h4 class="white"> <i class="fa fa-check-circle-o"> </i> <?php echo
htmlspecialchars($row['country']); ?> </h4>
<h4 class="white"> <i class="fa fa-twitter"> </i> <?php echo
htmlspecialchars($row['city']); ?> </h4>
<h4 class="white"> <i class="fa fa-twitter"> </i> <?php echo
htmlspecialchars($row['email']); ?> </h4>
<h4 class="white"> <i class="fa fa-twitter"> </i> <?php echo
htmlspecialchars($row['phone']); ?> </h4>

        <?php endwhile; ?>                                <a type="button"
class="btn btn-labeled btn-info" href="user-info.php">
        <span class="btn-label"> <i class="fa fa-pencil"> </i> </span> Update <
        </div>

</div>

<div class="col-md-6 no-pad">
<div class="user-image">

        <?php while ($row = $z->fetch()): ?>
<a href='profile.php'>  </a>

        <?php endwhile; ?>

</div>
</div>

```


</div>

<div class="row overview">

<div class="col-md-4 user-pad text-center">

<h3>FOLLOWERS</h3>

<h4>2,784</h4>

</div>

<div class="col-md-4 user-pad text-center">

<h3>FOLLOWING</h3>

<h4>456</h4>

</div>

<div class="col-md-4 user-pad text-center">

<h3>APPRECIATIONS</h3>

<h4>4,901</h4>

</div>

</div>

</div>

<div class="col-md-1 user-menu-btns">

<div class="btn-group-vertical square" id="responsive">

<i class="fa fa-bell-o fa-3x"> </i>

<i class="fa fa-envelope-o fa-3x"> </i>

<i class="fa fa-laptop fa-3x"> </i>

<i class="fa fa-cloud-upload fa-3x"> </i>

</div>

</div>

<div class="col-md-4 user-menu user-pad">

<div class="user-menu-content active">

<h3>

Recent Interactions

</h3>

<ul class="user-menu-list">

<h4> <i class="fa fa-user coral"> </i> Roselynn Smith followed you.</h4>

<h4> <i class="fa fa-heart-o coral"> </i> Jonathan Hawkins followed you.</h4>

<h4> <i class="fa fa-paper-plane-o coral"> </i> Gracie Jenkins followed you.</h4>


```
<button type="button" class="btn btn-labeled btn-success" href="#">
<span class="btn-label"> <i class="fa fa-bell-o"> </i> </span>View all
activity</button>
```

```
</li>
```

```
</ul>
```

```
</div>
```

```
<div class="user-menu-content">
```

```
<h3>
```

```
  Your Inbox
```

```
</h3>
```

```
<ul class="user-menu-list">
```

```
<li>
```

```
<h4>From Roselyn Smith <small class="coral"> <strong>NEW</strong> <i class="fa
fa-clock-o"> </i> 7:42 A.M.</small> </h4>
```

```
</li>
```

```
<li>
```

```
<h4>From Jonathan Hawkins <small class="coral"> <i class="fa fa-clock-o"> </i>
10:42 A.M.</small> </h4>
```

```
</li>
```

```
<li>
```

```
<h4>From Georgia Jennings <small class="coral"> <i class="fa fa-clock-o"> </i>
10:42 A.M.</small> </h4>
```

```
</li>
```

```
<li>
```

```
<button type="button" class="btn btn-labeled btn-danger" href="#">
<span class="btn-label"> <i class="fa fa-envelope-o"> </i> </span>View All
Messages</button>
</li>
</ul>
</div>
<div class="user-menu-content">
<h3>
    Trending
</h3>
<div class="row">
<div class="col-md-6">
<div class="view">
<div class="caption">
<p>47LabsDesign</p>
<a href="" rel="tooltip" title="Appreciate"> <span class="fa fa-heart-o
fa-2x"> </span> </a>
<a href="" rel="tooltip" title="View"> <span class="fa fa-search fa-2x"> </span> </a>

</div>

</div>
<div class="info">
```

```
<p class="small" style="text-overflow: ellipsis">An Awesome Title</p>
<p class="small coral text-right"><i class="fa fa-clock-o"></i> Posted Today | 10:42
A.M.</small>
</div>
<div class="stats turqbg">
<span class="fa fa-heart-o"><strong>47</strong></span>
<span class="fa fa-eye pull-right"><strong>137</strong></span>
</div>
</div>
<div class="col-md-6">
<div class="view">
<div class="caption">
<p>47LabsDesign</p>
<a href="" rel="tooltip" title="Appreciate"><span class="fa fa-heart-o
fa-2x"></span></a>
<a href="" rel="tooltip" title="View"><span class="fa fa-search fa-2x"></span></a>
</div>


</div>
<div class="info">
<p class="small" style="text-overflow: ellipsis">An Awesome Title</p>
<p class="small coral text-right"><i class="fa fa-clock-o"></i> Posted Today | 10:42
```

A.M.</small>

</div>

<div class="stats turqbg">

 47

 137

</div>

<div class="user-menu-content">

<h2 class="text-center">

About

</h2>

<center> <i class="fa fa-cloud-upload fa-4x"> </i> </center>

<div class="share-links">

</div>

</div>

</div>

<center> <button type="button" class="btn btn-lg btn-labeled btn-success"

href="#" style="margin-bottom: 15px;">

 <i class="fa fa-bell-o"> </i> A FINISHED PROJECT

</button> </center>

<center> <button type="button" class="btn btn-lg btn-labeled btn-warning"

href="#">

 <i class="fa fa-bell-o"> </i> A WORK IN PROGRESS

</button> </center>

</div>

</div>

</div>

</div>

</div>

</br>

<hr style="border:solid 1px">

<div id="container" style="padding-left:20px;margin-top:40px">

<h1>My Posts(<?php echo \$pc;?>)</h1>

<div id="columns">

<?php while (\$row = \$q->fetch()): ?>

<figure>

 <?php echo htmlspecialchars(\$row['user_name']); ?
>

<figcaption><p style="font-size:15px" ><?php echo
htmlspecialchars(\$row['body']); ?></p></figcaption>

 </figure>

 <?php endwhile; ?>

<small>Art © Claire Hummel</small>

 </div>

</div>

create-account.php(Registration page):

```
<?php
include('classes/DB.php');
if (isset($_POST['createaccount'])) {
    $username = $_POST['username'];
    $password = $_POST['password'];
    $email = $_POST['email'];

    $pdo = new PDO('mysql:host=127.0.0.1; dbname=socialnetwork;','root','kartik');
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    if ( empty($username) || empty($email) || empty($password) )
    {
        echo "Complete all fields</br><hr>";
    }

    if(!filter_var($email, FILTER_VALIDATE_EMAIL))
    {
        echo $emailvalid = "Enter a valid email</br><hr>";
    }

    if (strlen($password) <= 6){

        echo $passlength = "Choose a password longer then 6 character</br><hr>";
    }
}
```



```
$query = "SELECT * FROM users WHERE username = :username" ;
```

```
$statement = $pdo->prepare($query);
```

```
$pdoExec = $statement->execute(array(":username"=>$username));
```

```
if($pdoExec)
```

```
{
```

```
    if($statement->rowCount()>0)
```

```
    {
```

```
        echo $validus = "username already exists, select another
```

```
username</br><hr>";
```

```
    }
```

```
}
```

```
$query = "SELECT * FROM users WHERE email = :email" ;
```

```
$statement = $pdo->prepare($query);
```

```
$pdoExe = $statement->execute(array(":email"=>$email));
```

```
if($pdoExe)
```

```
{
```

```
    if($statement->rowCount()>0)
```

```
    {
```

```
        echo $validem = "Email-id already registered";
```

```
    }
```

```

    }

    if(empty($passmatch) && empty($emailvalid) && empty($passlength) &&
empty($validus) && empty($validem)) {

        $sql = "INSERT INTO users (username, password,email) VALUES
('$username','$password','$email')";

        $pdo->exec($sql);

        $query = "INSERT INTO users_info (username) VALUES ('$username')";

        $pdo->exec($query);
        echo "Success!";
        header('Location: login.php');
    }
}if(isset($_POST["Login"]))    {
    header('Location: login.    } ?>

<head> <style>
<?php include 'create-account.css'; ?>
</style>
</head>
<div class="container">
<h1 style="padding-left:80px; font-size:45px; font-family:Segoe
Script;">Register</h1>
<hr style="width:90%">

```

```
<form action="create-account.php" method="post">
<input type="text" name="username" value="" placeholder="Username ..." > </input>
<input type="password" name="password" value=""
placeholder="Password ..." > </input>
<input type="email" name="email" value=""
placeholder="someone@somesite.com" > </input>
<br>
<p>By clicking Register, you agree to our <a href="">Terms</a> and that you have
read our <a href="">Data use policy</a>, including our T&C. </p>
<input type="submit" name="createaccount" value="Register" class="button
button1">
</br> </br>
<br>
<hr style="width:90%">
<p>If you already have an account, then<input type="submit" name="Login"
value="Login" class="button button1"> </p>
</form>
</div>
```

login.php(login page)

```
$connect = new PDO('mysql:host=127.0.0.1;
dbname=socialnetwork;','root','kartik');

$connect->setAttribute(PDO::ATTR_ERRMODE,
PDO::ERRMODE_EXCEPTION);
```

//Button click login

code_____

```
        if(isset($_POST["login"]))
        {
            if(empty($_POST["username"]) || empty($_POST["password"]))
            {
                $message = '<label>All fields are required</label>';
            }
            else
            {
                $query = "SELECT * FROM users WHERE username = :username AND
password = :password";

                $statement = $connect->prepare($query);
                $statement->execute( array('username' =>
$_POST["username"],'password' => $_POST["password"] ));

                $count = $statement->rowCount();
                if($count > 0)
```

```
{  
  
    $_SESSION["username"] = $_POST["username"];  
  
    echo $_SESSION["username"];  
    header('Location: index.php');  
  
}  
else  
{  
    $message = '<label>Wrong Data</label>';  
  
}  
  
}  
  
}
```

//If error occurs_____

```
catch(PDOException $error)
```

```
{  
    $message = $error->getMessage();  
}
```

//Register button

code_____

```
if(isset($_POST["Sign-up"]))
```

```
{  
    header('Location: create-account.php');  
}
```

?>

<head>

<style>

<?php include 'login.css'; ?>

<style><?php include 'div.css'; ?> </style>

</style>

<head>

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

<div>

<h1 style="padding-left:30px; font-size:35px; font-family:Segoe Script;">Login to
your account</h1>

<hr style="width:90%">

<input type="text" name="username" value="" placeholder="Username ..."> </br>

<input type="password" name="password" value="" placeholder="Password ..."> </

br>

<input type="submit" name="login" value="login" class="button button2">

</br>

<hr style="width:90%">

<p>If you are not registered yet then</p>

<input type="submit" name="Sign-up" value="Sign-up" class="button button2">

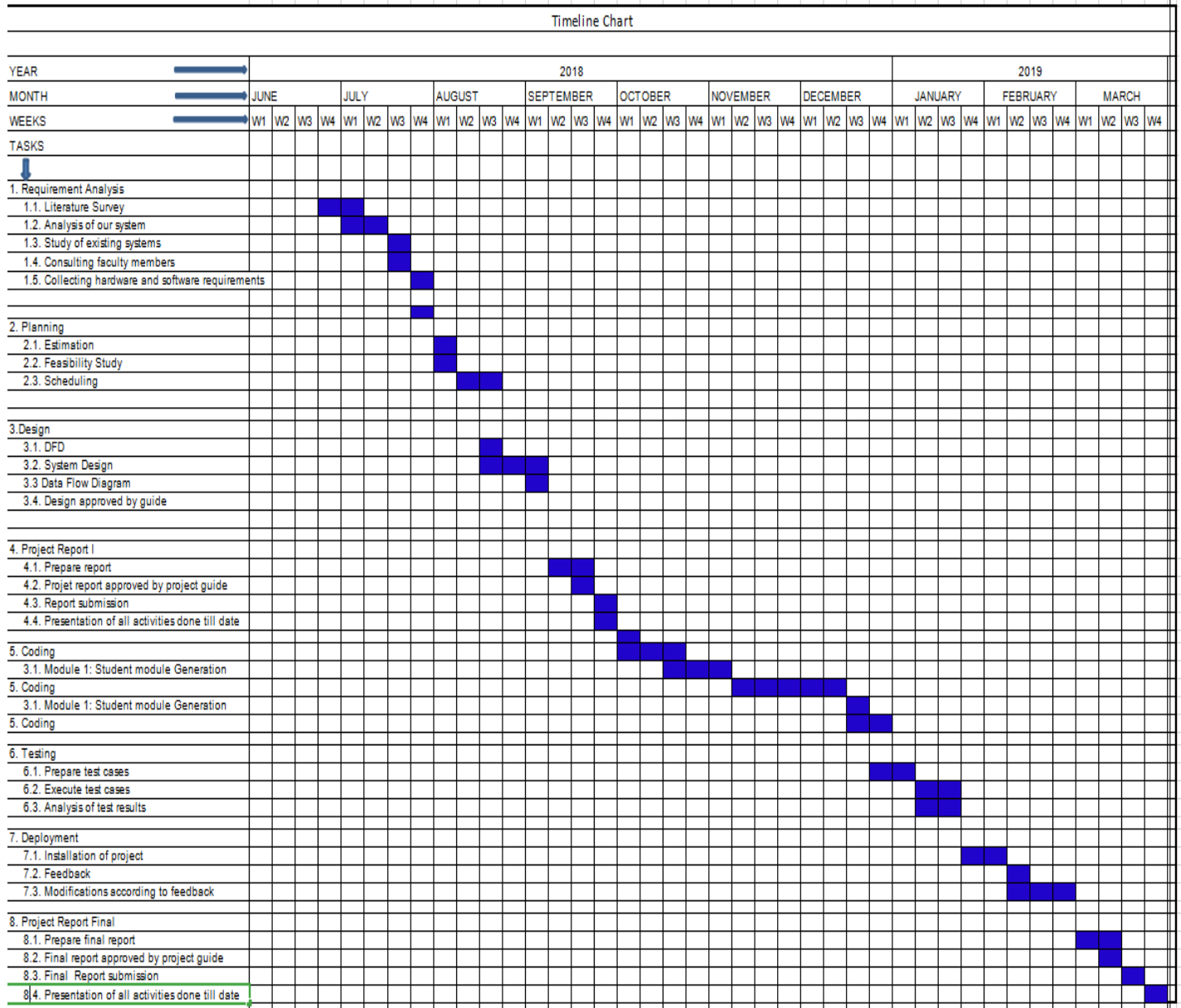
</br>

<?php

if(isset(\$message))

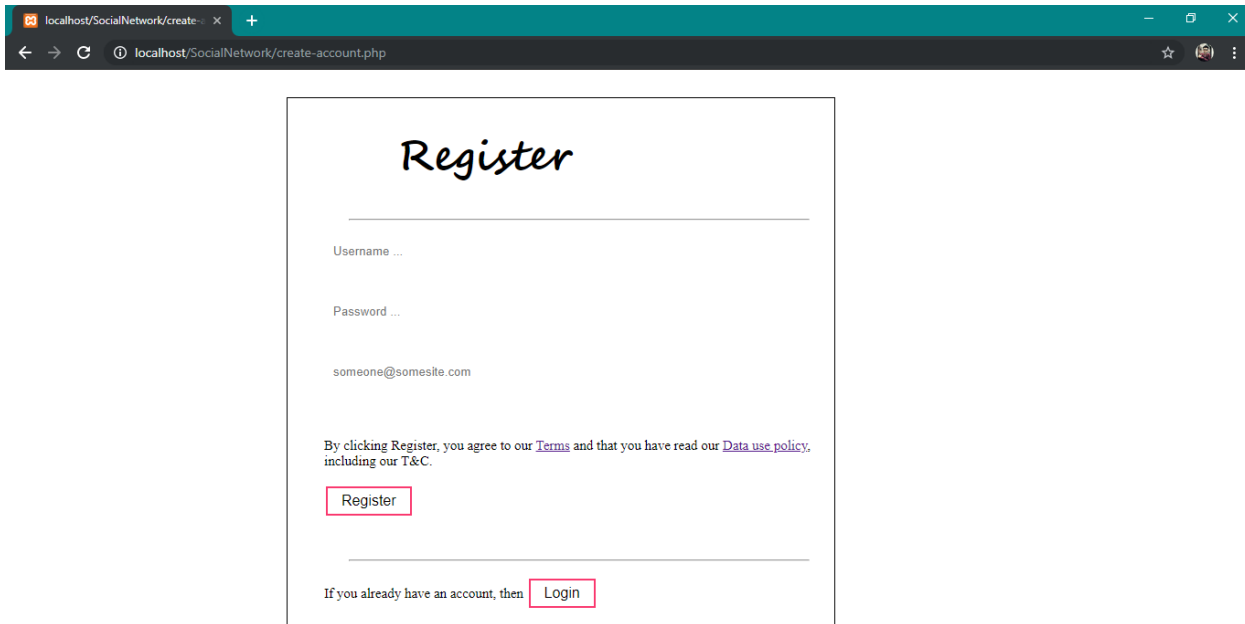
```
{  
    echo '<label class="text-danger">'.$message.'</label>';  
}  
?></form> </div>
```

4.1 Timeline Chart



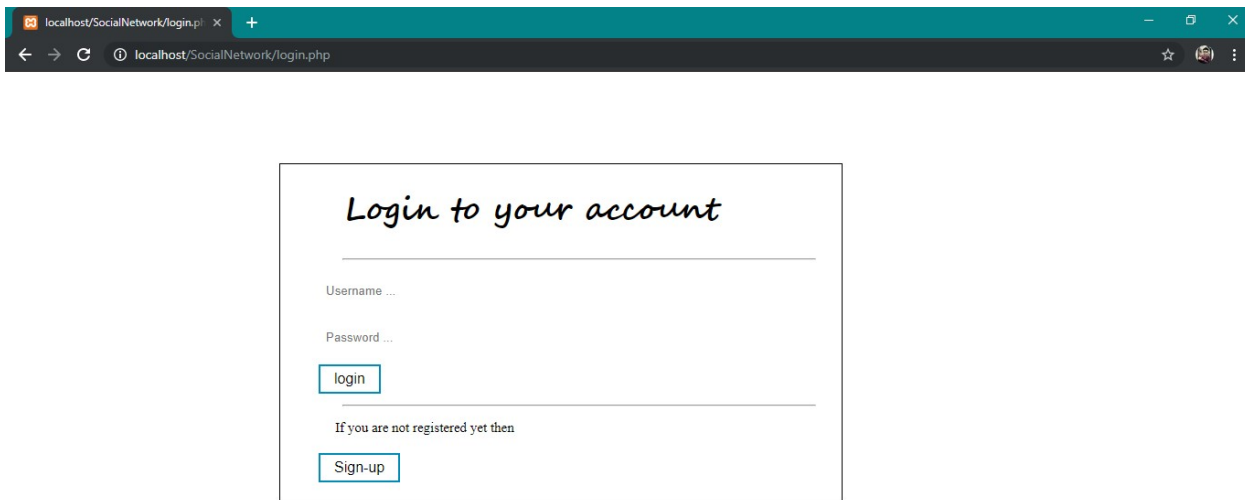
4.2 ScreenShots

Registration Page



The screenshot shows a web browser window with the address bar displaying 'localhost/SocialNetwork/create-account.php'. The page content is a registration form titled 'Register' in a large, black, cursive font. Below the title, there are three input fields: 'Username ...', 'Password ...', and an email field containing 'someone@somesite.com'. Below these fields, a line of text states: 'By clicking Register, you agree to our [Terms](#) and that you have read our [Data use policy](#), including our T&C.' Below this text is a red rectangular button labeled 'Register'. At the bottom of the form, there is a line of text: 'If you already have an account, then' followed by a red rectangular button labeled 'Login'.

Login Page



The screenshot shows a web browser window with the address bar displaying 'localhost/SocialNetwork/login.php'. The page content is a login form titled 'Login to your account' in a large, black, cursive font. Below the title, there are two input fields: 'Username ...' and 'Password ...'. Below the password field is a blue rectangular button labeled 'login'. Below this button, there is a line of text: 'If you are not registered yet then' followed by a blue rectangular button labeled 'Sign-up'.

HomePage


Project

[Logout](#) [Profile](#) [Settings](#)

Create a new Post, Share your content with the world..... [Click here](#)


News Feed

kartik




Posted at: 2019-10-02 17:32:03

kartik



Posted at: 2019-10-02 17:30:13


kartik



Posted at: 2019-10-02 17:30:13


HomePage(NewsFeed)

localhost/SocialNetwork/index.php




HANDMADE CLASSIC
Posted at: 2019-10-02 19:07:05

guitarworld




Collection
Posted at: 2019-10-02 19:06:40

kartik




Posted at: 2019-10-02 17:39:37



Posted at: 2019-10-02 17:30:13

memes



New meme... Hahahah
Posted at: 2019-09-30 00:48:09

MyProfile Page

Project

Create a new Post, Share your content with the world...Click here.

Welcome back,kartik
India
Navi mumbai
kartik.parmar.7500@gmail.com
9594898030

FOLLOWERS
0

FOLLOWING
0

APPRECIATIONS
0

Recent Interactions

Roselynn Smith followed you.

Jonathan Hawkins followed you.

Gracie Jenkins followed you.

Search Page

Your search results

[kartikspkar.par@gmail.com](#)

[kartikkartik.parmar.7500@gmail.com](#)

Image Upload page



5. TESTING

- **System Testing**

System testing is a critical element of quality assurance and represents the ultimate review of analysis, design and coding. Test case design focuses on a set of techniques for the creation of test because that meet overall testing objective. When a system is developed it is hoped that it performs properly. The main purpose of testing an information system is to find the errors and correct them. The scope of system testing should include both manual and computerized operations. System testing is comprehensive evaluation of the programs, manual procedures, computer operations and controls.

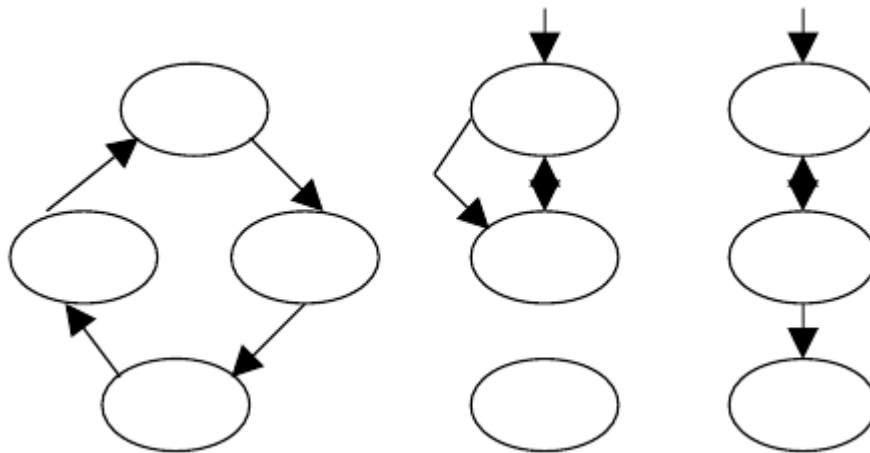
System testing is the process of checking whether the developed system is working according to the objective and requirement. All testing is to be conducted in accordance to the test conditions specified earlier. This will ensure that the test coverage meets the requirements and that testing is done in a systematic manner. The process of analyzing the software item to detect the differences between existing or required condition and evaluate the features of the software items. The thorough testing of the system before release of the software needs to be done vide the various test cases and modes so that the software becomes devoid of bugs and uses minimum space requirements as well as minimum time to perform. The test cases were selected beforehand with expected results defined and actual results recorded for comparison.

The selection of test cases is done vide -White Box Testing? technique to check the internal programming logic and efficiency and vide ?Black Box Testing? technique to check software requirement fulfillment with intension of finding maximum number of errors with minimum effort and time. Although test cases are a design by considering the cyclomatic complexity, conditional test, still the software code is not in its optional form, as all other possible alternative parts in the software are not considered. At the integration level, the software will be passing to the third party tests which would further enhance the software optimality and efficiency.

- **TEST DATA IMPLEMENTATION AND THEIR RESULT ON:**

The Quality and standardization of the software /application package depends truly on the various predefined testing norms and on the performances of the software over those norms. There are various standards existing in the software industry the engineered end product strives to achieve viz. ISO 9002 SEI CMM Level5 etc. These standards are achieved only when the concerns of tarefuls the tests as per the respective testing norms predefined in them vide the various test cases and parameters using the CASE topologies. Generally, software is tested both on a stand-alone mode as well after integrating all the modules in the system provide different available testing methods/norms.

The following Flow Graph methodology was used while testing the software:



Here each circle represents one or more non branching procedural language or source code statements in Flow Graph. While performing Condition Testing Domain Testing methodology was selected. While performing Loop Testing simple loops, concatenated loops, nested and unstructured loops were tested thoroughly.

• **TEST CHARACTERS :**

1. A good test has a high probability of finding an error.
2. A good test is not redundant .
- 3 .A good test should be - best of breed ? .
4. A good test should be neither too simple nor too complex.

- **BLACK BOX TESTING:**

The method of Black Box Testing is used by the software engineer to derive the required results of the test cases:

- 1.Black Box Testing alludes to test that are conducted at the software interface.
- 2.A Black Box Test examines some fundamental aspect of a system with little regard for the internal logic structure of the software.
- 3.A limited number of important logical paths can be selected and exercised.
- 4.Important data structure can be probed for validity.

Black box testing was performed to find errors in the following categories:-

- Incorrect or missing functions
- Graphics error.
- Errors in data in binary format.
- Error in data in integer format.
- File error.
- Pointer error.
- Memory access error.
- Variable error .
- Performance error

WHITE BOX TESTING:

White Box Testing is sometimes called Glass Box Testing. Using White Box Testing methods the software engineer can derive the following test cases:

- Guarantee that all independent paths within a module have been exercised at least once.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to ensure the validity.

In White Box Testing efforts were made to handle the following:-

- Number of input parameters equal to number of arguments.
- Parameters and arguments attributes match.
- Number of arguments transmitted is called modules equal to attributes of parameters.
- Unit system of argument transmitted is called module's equal unit system of parameter.
- Number of attributes and order of arguments to build in functions correct.
- Any references to parameters not associated to build in functions correct.
- Input only arguments altered.
- Global variable definition consistent across module.
- Files attributes correct.
- Format specifications matches I/O specification.

- Files opened before use.
- File closed while working is going on.
- I/O errors handled.
- Any textual errors in output information.

UNIT TESTING:

The unit testing is performed to test the validity of the individual units. This is done in the coding phase with the interactive testing. Thus it itself constitutes a majority of functionality test for each logical unit.

INTEGRITY TESTING:

When all the development of all the units or modules is completed and integrated the integrity test phase is started. In this phase the interface between the modules are tested .

5.1 TEST CASES

Login Page

| Sr No. | Input Values | Test case | Conditional being checked | Result |
|--------|--------------|-------------------|------------------------------|------------|
| 1 | username | Empty | All fields are required | Successful |
| 2 | password | Empty | All fields are required | Successful |
| 3 | username | If wrong username | Invalid username or password | Successful |
| 4 | password | If wrong password | Invalid username or password | Successful |

Registration Page

| Sl No. | Input Values | Test case | Conditional being checked | Result |
|--------|--------------|------------------------|--|------------|
| 1 | username | Empty | All fields are required. | Successful |
| 2 | password | Empty | All fields are required. | Successful |
| 3 | email | Empty | All fields are required. | Successful |
| 4 | password | Length | Password must be more than 6 characters. | Successful |
| 5 | username | Already exists or not. | Username already exists. | Successful |
| 6 | email | Already exists or not. | Email already Registered. | Successful |
| 7 | email | Invalid email | Enter a valid email-id | Successful |

Post Upload:

| Sr No. | Input Values | Test case | Conditional being checked | Result |
|---------------|---------------------|-------------------|--|---------------|
| 1 | File-Uploader | Null | Only JPG, JPEG, PNG & GIF files are allowed. | Successful |
| 2 | File-Uploader | Invalid File Type | Only JPG, JPEG, PNG & GIF files are allowed. | Successful |
| 3 | Caption | Length | Caption must be under 60 characters | Successful |

Update Profile-pic:

| Sr No. | Input Values | Test case | Conditional being checked | Result |
|---------------|---------------------|-------------------|--|---------------|
| 1 | File-
Uploader | Invalid File Type | Only JPG, JPEG, PNG & GIF files are allowed. | Successful |

Search:

| Sr No. | Input Values | Test case | Conditional being checked | Result |
|---------------|---------------------|------------------|----------------------------------|---------------|
| 1 | SearchBox | Exists or not | No results found:"XYZ" | Successful |
| 2 | SearchBox | Null | Search for users, Enter Username | Successful |

6. CONCLUSION

While developing the system a conscious effort has been made to create and develop a software package, making use of available tools, techniques and resources – that would generate a proper system for ONLINE SOCIAL NETWORKING.

While making the system, an eye has been kept on making it as user-friendly. As such one may hope that the system will be acceptable to any user and will adequately meet his/her needs.

As in the case of any system development process where there are a number of shortcomings, there have been some shortcomings in the development of this system also. There are some of the areas of improvement which couldn't be implemented due to time constraints. One such feature was online chat where users can chat with other users through this website.

7 .REFERENCES

[1]Author Mario Lurig,s :

PHP Reference: Beginner to Intermediate PHP5 Ebook(PDF)

[2] PHP The Right Way

<<https://phptherightway.com/>>

[3] PHP Pdo queries

<<https://phpdelusions.net/pdo>>

[4]PHP MySQL PDO Database Class

<<https://codereview.stackexchange.com/questions/156968/php-mysql-pdodatabase-class>>

[5] Howcode Youtube channel php Tutorials for Social-Networking Website

<https://www.youtube.com/playlistlist=PLBOh8f9FoHHhRk0Fyus5MMeBsQ_qwIAzG>