**FREEDOM**

**(free to chat and discuss on messages )**

**The Social networking site**

****

**-**

**Department of Computer and Information Technology**

**Jaypee Institute of Information Technology, noida**

**Mentor:**

**Prashant Kaushik**

**By:**

**SUMIT BANSAL (12103530)**

**VRISHTI GAHLAUT (12103528)**

**DHRUV SINGH (12103533)**

**HAMZA KHAN (12103540)**

**Abstract**

A social networking service (also social networking site or SNS) is a platform to build social networks or social relations among people who share interests, activities, backgrounds or real-life connections. A social network service consists of a representation of each user (often a profile), his or her social links, and a variety of additional services. Social network sites are web-based services that allow individuals to create a public profile, to create a list of users with whom to share connections, and view and cross the connections within the system.

To design a social networking site through which one user can be connected with other, view the information and share it. With an android application to chat on the go!!

**TABLE OF CONTENTS**

**1. INTRODUCTION**

**2. LITERATURE REVIEW**

**3. PROJECT DESIGN**

**3.1 Problem Statement**

***3.1.1 Database Design***

**3.2PROCESS MODEL**

***3.2.1 Agile Process Model***…………………………………………………………

**3.3Unified Modelling Language**

**4. Tools/Languages Used**

**4.1. Open Source Tools**

**4.2 Integrating the Website and Database**

**4.3. Server Side Processing**

**4.4.Client Side Processing**

**5. WEB Based Development**

**5.1. The Social Network**

**5.2. List of Project Features.**

**5.3. List of Features Implemented**

**5.3. List of Features to be implemented**

**6. Functional & Non-Functional Requirements**

**7.Snapshots**

**8. Conclusion**

­

**9. Bibliography**

**1. Introduction**

A social networking service (also social networking site or SNS) is a platform to build social networks or social relations among people who share interests, activities, backgrounds or real-life connections. A social network service consists of a representation of each user (often a profile), his or her social links, and a variety of additional services. Social network sites are web-based services that allow individuals to create a public profile, to create a list of users with whom to share connections, and view and cross the connections within the system.

Development of a social networking site which can fulfil requirements like:

1. Send/receive friend request.

2. Input personal information.

3. View the personal information.

4. View profile picture.

5. Crawl images and videos.

6. Upload pictures.

7. play games .

8. chat with friends.

It will be a platform where friends can send and receive messages. Also an android application for chatting will be available for instant messaging.

Android Application :

1. Bluetooth chat

2. Free messages

3. Email

4. Play games

5. Share images and selfies

**2. Literature Review**

A social networking service (also social networking site or SNS) is a platform to build social networks or social relations among people who share interests, activities, backgrounds or real-life connections. A social network service consists of a representation of each user (often a profile), his or her social links, and a variety of additional services. Social network sites are web-based services that allow individuals to create a public profile, to create a list of users with whom to share connections, and view and cross the connections within the system.[1] Most social network services are web-based and provide means for users to interact over the Internet, such as e-mail and instant messaging. Social network sites are varied and they incorporate new information and communication tools such as mobile connectivity, photo/video/sharing and blogging.[2] Online community services are sometimes considered as a social network 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, pictures, posts, activities, events, interests with people in their network.

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 self-description pages), and a recommendation system linked to trust.

**3. Project Design**

In order to design a web site, the relational database must be designed first. Conceptual design can be divided into two parts: The **data model** and the **process model**. The data model focuses on what data should be stored in the database while the process model deals with how the data is processed. To put this in the context of the relational database, the data model is used to design the relational tables. The process model is used to design the queries that will access and perform operations on those tables.

**3.1 Problem Statement**

Development of a social networking site which can fulfil requirements like:

1. Send/receive friend request.

2. Input personal information.

3. View the personal information.

4. View profile picture.

5. Crawl images and videos.

6. Upload pictures.

7. play games .

8. chat with friends.

It will be a platform where friends can send and receive messages. Also an android application for chatting will be available for instant messaging.

Android Application :

1. Bluetooth chat

2. Free messages

3. Email

4. Play games

5. Share images and videos

**GANT CHART**

A **Gantt chart** is a type of bar chart, developed by Henry Gantt in the 1910s, that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the work breakdown structure of the project. Modern Gantt charts also show the dependency (i.e., precedence network) relationships between activities. Gantt charts can be used to show current schedule status using percent-complete shadings and a vertical "TODAY" line as shown here.

Although now regarded as a common charting technique, Gantt charts were considered revolutionary when first introduced. This chart is also used in information technology to represent data that have been collected.

**HISTORICAL DEVELOPMENT:**

The first known tool of this type was developed in 1896 by Karol Adamiecki, who called it a harmonogram. Adamiecki published his chart in 1931, however, only in Polish, which limited both its adoption and recognition of his authorship. The chart is named after Henry Gantt (1861–1919), who designed his chart around the years 1910–1915.

One of the first major applications of Gantt charts was by the United States during World War I, at the instigation of General William Crozier.

In the 1980s, personal computers allowed for widespread creation of complex and elaborate Gantt charts. The first desktop applications were intended mainly for project managers and project schedulers. With the advent of the internet and increased collaboration over networks at the end of the 1990s, Gantt charts became a common feature of web-based applications, including collaborative groupware.

**FURTHER APPLICATIONS:**

Gantt charts can be used for scheduling generic resources, so as well as their use in project management, they can also be used in scheduling production processes and employee rostering. In the latter context, they may also be known as time bar schedules.

There are many computer applications supporting the use of Gantt charts for employee scheduling. Gantt charts can be used to track shifts or tasks and also vacations or other types of out-of-office time.

**3.1.1 Database Design**

In Database , we store :

1. user name and password
2. All the user personal information
3. pictures of user
4. Freedom gallery
5. Uploaded picture of users
6. News feed
7. Total visitors
8. Photo like
9. Chat
10. User’s profile
11. Change theme
12. News feed
13. Top trends
14. Notes

**3.2. Process Model**

A Process Model tells us about how the data is processed and how the data flows from one table to another to gather the required information. This model consists of the Functional Decomposition Diagram and Data Flow Diagram.

**3.2.1Process Model**

A process model is chosen based on the nature of the project and application, the methods and tools to be used, and the controls and deliverables that are required. Our project is implemented using method

Extreme programming in Agile process model.

Agile Model:

Agile development model is also a type of Incremental. Software is developed in incremental, rapid cycles. This results in small incremental releases with each release building on previous functionality. Each release is thoroughly [tested](http://istqbexamcertification.com/why-is-testing-necessary/) to ensure [software quality](http://istqbexamcertification.com/what-is-software-quality/) is maintained. It is used for time critical applications.

* Agile Methods:
* Extreme Programming
* SCRUM
* OpenUP
* LEAN Development
* Crystal
* Test Driven Development
* Extreme Programming: Extreme Programming (XP) takes an ‘extreme’ approach to iterative development.

1. New versions may be built several times per day;
2. Increments are delivered to customers every 2 weeks;
3. All tests must be run for every build and the build is only if tests run successfully.
4. There are some steps which are followed in extreme programming:

* Incremental Planning
* Simple and Effective Design
* Test first development
* Refactoring
* Pair Programming
* Continuous Integration
* **Incremental Planning:**

Requirements are recorded on Story Cards and the Stories to be included in a release are determined by the time available and their relative priority. In our project we used incremental planning. We divide our project requirements as story card. Stories in our project are

1. Web designing: In web designing first we framed a basic HTML page on which we embedded CSS then following by CSS we developed server side scripting in PHP and then client side scripting same as in PHP
2. Parallax
3. Collaborative designing

* **Simple and Effective Design:**

Enough design is carried out to meet the current requirements and no more. After the planning phase the next phase is the Design phase where application is designed according to requirements. This phase is very important phase because it will structure the layout of our application. Our website doesn’t have extraneous information. Website have simple typography so it load faster and also have smaller file sizes which means it takes less server space and bandwidth. Designing of our website follow the optimize number of HTTP calls so that website load and respond faster.

* **Test first development:**

An automated unit test framework is used to write tests for a new piece of functionality before that functionality itself is implemented.

1) Writing tests before code clarifies the requirements to be implemented.

2) Tests are written as programs rather than data so that they can be executed automatically. The test includes a check that it has executed correctly.

3) All previous and new tests are automatically run when new functionality is added. Thus checking that the new functionality has not introduced errors.

* **Refactoring:**

All developers are expected to refactor the code continuously as soon as possible code improvements are found. This keeps the code simple and maintainable.

1) Refactoring is the process of code improvement where code is re-organised and rewritten to make it more efficient, easier to understand, etc.

2) Refactoring is required because frequent releases mean that code is developed incrementally and therefore tends to become messy.

3) Refactoring should not change the functionality of the system.

4) Automated testing simplifies refactoring as you can see if the changed code still runs the tests successfully.

**In our project we refactor the code at each stage of development because** harder it is to see the design in the code, the harder it is to preserve it, and the more rapidly it decays. Regular refactoring helps code retain its shape.

We applied different Techniques that allow for more abstraction such as:

1. Replace type-checking code with State/Strategy.
2. Create more general types to allow for more code sharing.

* **Pair Programming:**

1) In XP, programmers work in pairs, sitting together to developcode.

2) This helps develop common ownership of code and spreads knowledge across the team.

3) It serves as an informal review process as each line of code is looked at by more than 1 person.

4) It encourages refactoring as the whole team can benefit from this.

5) Measurements suggest that development productivity with pair programming is similar to that of two people working independently. In our development we apply some strategy based on pair programming:

1. The task was something we are confident that we can complete in an hour or two.
2. The key to good pairing is to sync up very frequently—within seconds or a minute of noticing that we are out of sync. If we are spending five minutes (or more) out of sync, we might as well be coding solo, because it's the frequent re-sync'ing that creates the synergy of pairing.
3. Switch roles often—at least every half hour.
4. Rely on partner as well as support partner.

* **Continuous Integration:**

As soon as work on a task is complete it is integrated into the whole system. After any such integration, all the unit tests in the system must pass.

* 1. **Unified Modelling Language:**

Describing a system at a high level of abstraction.

* A model of the system
* Used for requirements and specifications
* It is an industry-standard graphical language for specifying, visualizing, constructing, and documenting the artifacts of software systems
* The Unified Modeling Language uses mostly graphical notations to express the Object Oriented analysis and design of software projects.
* Simplifies the complex process of software design.
* Use graphical notation to communicate more clearly than natural language (imprecise) and code (too detailed).
* Help acquire an overall view of a system.
* Unified Modeling Language is not dependent on any one language or technology.
* Unified Modeling Language moves us from fragmentation to standardization.
* Unified Modeling Language includes mostly five diagrams:
* Use case Diagram
* Class Diagram
* Sequence Diagram
* State Diagram
* Activity Diagram
  + 1. **Use Case Diagram:**
* Use cases serve as a technique for capturing the functional requirements of a system.
* Describes the typical interactions between the users of a system and the system itself, providing a narrative of how a system is used.
* A use case consists of a set of one or more scenarios tied together by a common user goal.
* A scenario is a sequence of steps describing an interaction between a user and a system; some scenarios describe successful interaction; others describe failure or errors
* Users are referred to as actors; an actor is a role that carries out a use case
* An actor need not always be a person; it can also be an external system that is either automated or manual.
* A use case diagram is like a graphical table of contents of the use cases for a system
* It shows the use cases, the actors, and the relationships between them.
* Use cases represent an external view of the system; consequently, they have no correlation to the classes in the system

-They can serve as a starting point for writing software validation test cases.

* Use case diagram core components:
* **Actor**: A role that a user plays with respect to the system, including human users and other systems. An external system that needs some information from the current system.
* **Use case:** A set of scenarios that describing an interaction between a user and a system, including alternatives.
* **System boundary**: rectangle diagram representing the boundary between the actors and the system.
* **Association**: communication between an actor and a use case; Represented by a solid line.

* **Include**: a dotted line labeled <<include>> beginning at base use case and ending with an arrows pointing to the include use case. Include relationship occurs when a chunk of behavior is similar across more than one use case. Use “include” instead of copying the description of that behavior.
* **Extend**: a dotted line labeled <<extend>> with an arrow toward the base case. The extending use case may add behavior to the base use case.

Social networking site

**New user**

Administrator

New User

Registered User

# **Tools/Languages Used:**

* JavaScript version 1.8.5
* jarallax library version: 0.2.1
* jQuery v1.7.1
* WAMP for windows 32 bit
  + - * + Apache 2.2.4 web server
        + MySQL 5.0.41
        + PHP 5.2.3
        + phpMyAdmin 4.2.10.1
  + HTML 5
  + CSS 3
  + Macromedia Dreamweaver
  + Notepad++
  + Sublime text

**4.1. Open source tools:**

* JS SLIDER:

It is used to slide images with different animations.

Source: <http://www.menucool.com>

* PHPMAILER:

PHPMailer is a PHP class for PHP (www.php.net) that provides a package of functions to send email. The two primary features are sending HTML Email and e-mails with attachments. PHPMailer supports nearly all possibilities to send email: mail(), Sendmail, qmail & direct to SMTP server. You can use any feature of SMTP-based e-mail, multiple recipients via to, CC, BCC, etc. In short: PHPMailer is an efficient way to send e-mail within PHP.

As you may know, it is simply to send mails with the PHP mail() function. So why use PHPMailer? Isn't it slower? Yes that's true, but PHPMailer makes it easy to send e-mail, makes it possible to attach files, send HTML e-mail, etc. With PHPMailer you can even use your own SMTP server and avoid Sendmail routines used by the mail() function on Unix platforms.

* GOOGLE MAPS API

Google Maps is a desktop and mobile web mapping service application and technology provided by Google, offering satellite imagery, street maps, and Street View perspectives, as well as functions such as a route planner for traveling by foot, car, bicycle (beta test), or with public transportation. Also supported are maps embedded on third-party websites via the Google Maps API, and a locator for urban businesses and other organizations in numerous countries around the world. Google Maps satellite images are not updated in real time; however, Google adds data to their Primary Database on a regular basis, and most of the images are no more than 3 years old.

The opt-in redesigned version of the desktop application has been available since 2013, alongside the "classic" (pre-2013) version. The redesigned version was met by user criticism regarding slowness, hiding some common functions, removing a scale bar, and lack of other features that include My Places and sharable customized links to parameterize split Street View and Map views.  It is possible to switch back to the old version.

Google Maps uses a close variant of the Mercator projection, and therefore cannot accurately show areas around the poles. A related product is Google Earth, a stand-alone program which offers more globe-viewing features, including showing polar areas.

Google Maps for mobile is the world's most popular app for smartphones, with over 54% of global smartphone owners using it at least once during the month of August 2013.

Google Maps has a wide array of APIs that let you embed the robust functionality and everyday usefulness of Google Maps into your own website and applications, and overlay your own data on top of them.

* YOUTUBE XML DATA

The YouTube Data API retrieves search results from a specially optimized search index. The index is designed to include new videos as quickly as possible while ensuring high performance even under heavy API server loads. Please note that there may be delays between the time that a video is uploaded or updated and the time the new information is included in video feeds. The following list provides expected data latencies for various API functions:

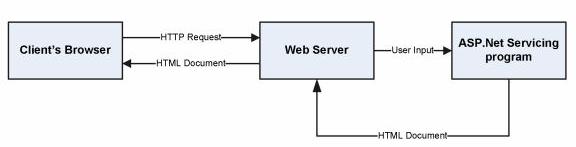
* Standard feeds are updated periodically, with the update frequency varying from feed to feed. Many feeds are updated every 30 to 40 minutes but other feeds – such as those that capture daily, weekly or monthly results – may only be updated hourly or even daily.
* Uploaded videos will be included in a user's public uploaded videos feed a few minutes after the upload completes and YouTube finishes processing the video.
* Uploaded videos will usually be available in search feeds within 30 minutes to two hours after the upload completes and YouTube finishes processing the video. However, this delay may be longer under heavy API server loads.
* Video metadata updates are typically reflected in search feeds within 30 minutes to two hours after the upload completes. However, this delay may be longer under heavy API server loads, and updates may take up to 24 hours to show up.
* Deleted videos will usually be removed from a user's public uploaded videos feed within 30 minutes to two hours after the upload completes. However, this delay may be longer under heavy API server loads.
* Statistics, such as the number of times a video has been viewed or the number of comments available for a video, are typically updated within 30 minutes to two hours after the upload completes. However, this delay may be longer under heavy API server loads.
* On an ongoing basis, statistics for a video are typically updated every 30 minutes to two hours. However, updates may occur less frequently under heavy server loads or for videos that are viewed very infrequently. In search feeds, updates to ratings and view counts could take as long as a couple of weeks for infrequently viewed videos.
* Events that are listed in activity feeds will usually be available in those feeds within 30 minutes of the times that they occur.

**4.2. Integrating the Website and Database**

**4.3. Server-side processing.**

Generally dynamic or data-driven Web pages use HTML forms to collect user inputs, submitting them to a Web server. A program running on the server processes the form inputs, dynamically composing a Web page reply. This program, which is called, servicing program, can be either a compiled executable program or a script interpreted into machine language each time it is run.

Compiled server programs: When a user submits HTML- form data for processing by a compiled server program, the Web Server invokes the servicing program. The servicing program is not part of the Web server but it is an independent executable program running on the Web server; it processes the user input, determines the action which must be taken, interacts with any external sources (Eg: database) and finally produces an HTML document and terminates. The Web server then sends the HTML document back to the user’s browser where it is displayed. Next figure shows the flow of HTTP request from the client to the Web server, which is sent to the servicing program. The program creates an HTML document to be sent to the client browser.



**Figure 23 Compiled server programs flowchart**

**4.4. Client-Side Processing.**

Client-side Web page processing is achievable through compiled programs downloaded, installed, and executed on the client workstation or by creating scripts with the HTML Web page commands interpreted by the client browser.

Downloading and running compiled programs on client workstations: When a user clicks a hyperlink on a Web page associated with a compiled client-side program, the user’s browser must have the ability to run the executable program file; this program interacts with the user, sending and retrieving data from a database server as needed.

Client-side scripts: In a client-side script, source code written in such languages as JavaScript is embedded in an HTML document, along with the static HTML text; it is placed within delimiter tags to indicate to the user’s browser that the text is code that must be interpreted. If the user’s browser is able to recognize and interpret the code, it is processed. If the browser is unable to recognize and interpret the code, it is displayed as text on the Web page.

Client-side scripts can also be used to create advanced Web page features, including: animations, calculations, playing sound and video, and image maps allowing users to move their cursors over an image and click to access different Web page links.

JavaScript is the most commonly used client-side scripting language and is supported by most browsers.

**5. Web Based Application Development**

The Web is built on the HyperText Transfer Protocol. HTTP is a client/server request/reply protocol that is *stateless*. That is, the protocol does not make any association between one transaction and another; e.g.: time since the last transaction, type or client involved in the last transaction, what data was exchanged between the client and the server. As far as HTTP is concerned, each transaction is a discrete event. But this is not what we want in a shopping cart application because we need to preserve the user’s shopping selection as they proceed with their purchase, in addition it is useful to have the access to their past purchase history and personal preferences.

Carrying information from one page to another can be achieved by several ways, such as Cookies, Session variables, Post variables, etc.

A cookie is a small file that has a maximum age, a domain and path of applicability, and a security specification. Any time a server sends a response to a client, it may include one or more Set-Cookie headers. When a client receives a Set-cookie header, it stores the content of the header and the cookie, for later use. In our application, every time the client selects an item to put in the shopping cart, the server can send a Set-Cookie whose content is the ID of the item, and whose domain and path of applicability are the URL of the order/payment page. Then, when the user goes to order and pay, the client will send the Cookie headers for each of the selected items. Upon receiving this request, the server can parse the supplied cookies and charge the user appropriately for the selected items. Cookies may also be used to identify the users.

However, cookies are very insecure to use since they are transmitted as plain text and the server has no control over how cookies are stored in at the client’s side. Another approach is based on a notion of session ID. These notions provide means for the server to track the requests of a client through a “session”, but unlike cookies, which are stored on the client, Session variables are stored on the Server. A session starts when a user logs in and ends when they log off from the website.

The Session object is used to store information about, or change settings for a user session. Variables stored in the Session object hold information about one single user, and are available to all pages in one application. Common information stored in session variables are name, id, and preferences. The server creates a new Session object for each new user, and destroys the Session object when the session expires.

In this project, the concept of session variables will be used for maintaining state information.

**5.1. The Social Network : FREEDOM**

The objective of this application is to provide the user an online website where they can chat with friends and view their information and photos.

Website consists of the following web pages:

1. Index.php
2. Album.php
3. Again.php
4. Features.php
5. Friend.php
6. Gallery.php
7. Game.php
8. Home1.php
9. Imagecrawl.php
10. Inbox.php
11. Indexgallery.php
12. Like.php
13. Like1.php
14. Like2.php
15. Profile.php
16. profileAdmin.php
17. registration1.php
18. reset.php
19. simple\_html\_dom.php
20. sub1.php
21. sub2.php
22. sub3.php
23. web.php
24. video.php
25. xor.php
26. about
27. chat
28. search
29. crawl.php
30. theme.php
31. twit4.php

**5.2. List of Project features:**

-main project

• Simple and attractive

• Secure login

• Make friends

• Login and registration form

• Password changing form

• User friendly interface

• Group Discussion

• Friends Activity display

• Contact information

• View friend’s Personal information

• Chat with friends

• Image crawl

• Video crawl

• search engine

• games

• upload photos

• freedom gallery and videos

• public and private chat

• Admin rights

• total visitors

• responsible website

• Top trends

• News Feed

• background video

• Change home theme

• Feedback form

• About us

-android application

• Simple and attractive

• three ways to send a message

• normal messaging free of cost

• Bluetooth chatting

• It’s 100% free.

• User friendly interface

• games

• send photos

**User stories**

Story 1. A customer connects to the site looking for friends:

1. The customer starts with about us main page.

2. The customer signs up.

3. The customer gives information.

4. The customer gets others information.

5. The customer decides if she wants to proceed with any friend request

Story 2. A customer connects to chat for free through mobile

1. The customer installs the application

2. The customer browsers through the menu and chooses to chat.

3. The customer chats and can send images.

4. The customer can even compose a mail .

5. He/She can play games for fun.

**Functional requirements based on processes**

**Customer management**

It's a common scenario that a new user registers using the front-end to be able to :

• Check if a username already exists in Database.

• Create new users in Database.

• Modify the information of a user in Database.

**User Interface Mock-ups:**

**Technical Requirements**

Web services allows applications to be integrated more rapidly and easily. These characteristics are ideal for connecting functions across heterogeneous systems.

The integration requirements are:

• The system makes calls to the Database web service whenever it requires a synchronization.

• The definition of these web services should not change from version to version to guarantee the integration with different Database versions.

**Non-Functional Requirements**

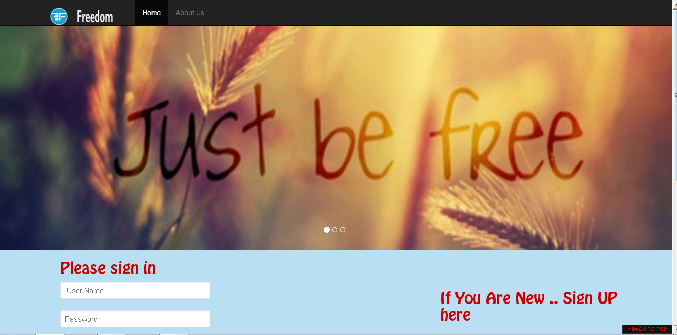
These are general requirements that impact in the technical decisions:

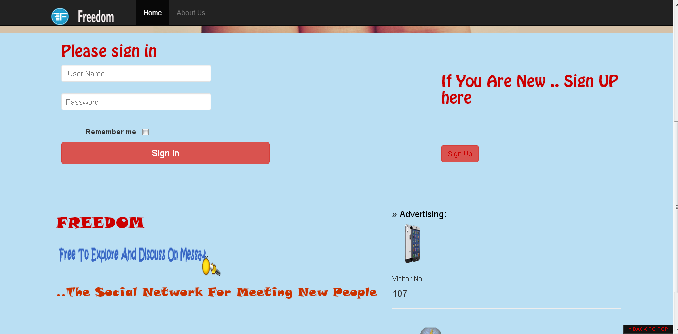
• The final system should scale correctly to support high traffic loads.

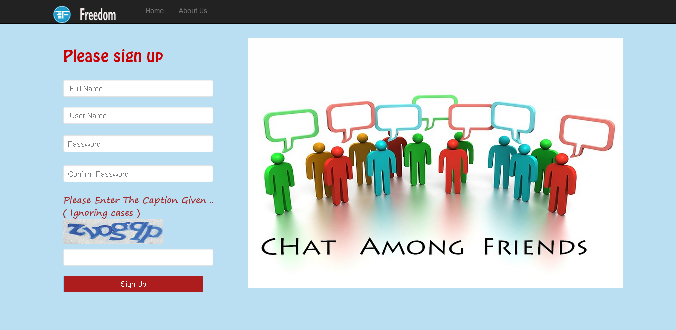
• Security is key since we are performing commercial translations and dealing with confidential data.

• It is important that all process within Database are automatic.

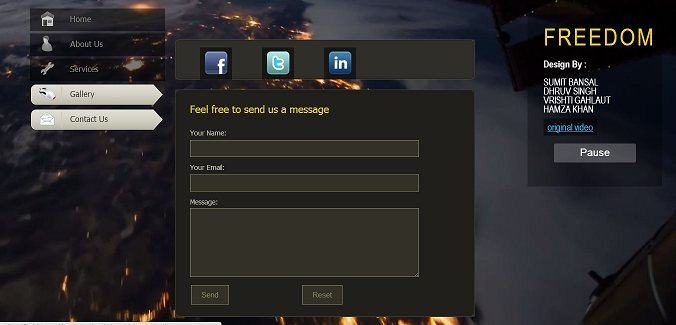
**7. SNAPSHOTS**

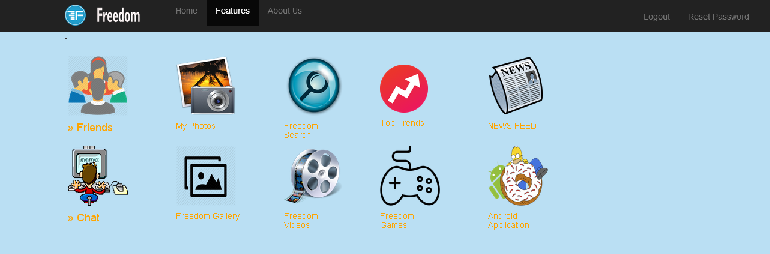
****

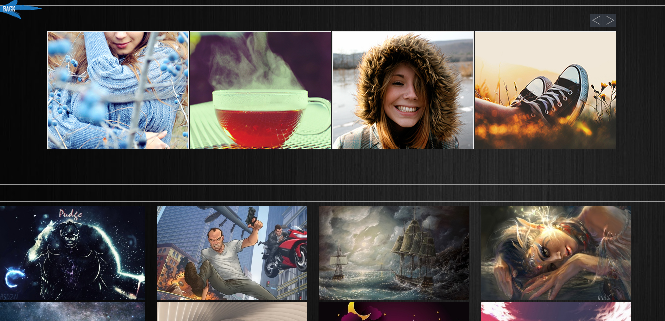
****

****

****

****

****

****

**8. Conclusion**

Social Networking website is a revolutionary idea with a very bright future with further scope for advancements. The opportunities provided from this medium are immense and many organisations are making use of this medium to better their practices. Organisations are no longer at the mercy of the media to advertise or convey their message. With the help of social networking they can advertise or communicate in a more efficient way. For example, Starbucks have started a very successful program in which a person from any part of the world can login to a website to write comments and discuss issues. Similarly people don’t have to rely on newspapers or TV to get their daily dose of news it can all be obtained from a social networking site. People can follow or get information from any part of the world. For example Twitter allows a user to follow anything from airline timing to the next breaking news from China. It is even used by politicians to get their message across.

**9. Bibliography**

1. http://encyclopedia.laborlawtalk.com/MYSQLfor information on MYSQL
2. http://aspnet.4guysfromrolla.com/articles/020404-1.phpfor relationship between MYSQL and PHP.
3. http://216.15.201.66/dpec/course.htm?fullpg=http%3A//216.15.201.66/dpec/cours es/wac312/wah006.htm&acro=wac312for security authentication in PHP
4. http://samples.gotdotnet.com/quickstart/aspplus/doc/mtstransactions.phpforinformation on Transactions in PHP.
5. http://www.x-cart.com/articles/design\_development.htmlfor online customer behaviour.
6. http://aspnet.4guysfromrolla.com/articles/011404-1.phpfor relation between MYSQL and PHP.
7. [http://www.informatik.uni-bremen.de/uniform/gdpa\_d/methods/m- fctd.htm](http://www.informatik.uni-bremen.de/uniform/gdpa_d/methods/m-%20fctd.htm) for definition of Functional Decomposition.
8. http://www.agilemodeling.com/artifacts/dataFlowDiagram.htmfor definition of Data Flow Diagram.
9. http://www.startvbdotnet.com/ado/default.phpfor information on PHP
10. http://mypage.iusb.edu/~hhakimza/505/index.htmlfor PHP objects.
11. http://msdn.microsoft.comfor PHP objects.