

2015

Collaboration Portal for Researchers at Eastern Illinois University

Fatemeh Khastkhodaardekani

Eastern Illinois University

This research is a product of the graduate program in [Technology](#) at Eastern Illinois University. [Find out more](#) about the program.

Recommended Citation

Khastkhodaardekani, Fatemeh, "Collaboration Portal for Researchers at Eastern Illinois University" (2015). *Masters Theses*. 2362.
<https://thekeep.eiu.edu/theses/2362>

This is brought to you for free and open access by the Student Theses & Publications at The Keep. It has been accepted for inclusion in Masters Theses by an authorized administrator of The Keep. For more information, please contact tabruns@eiu.edu.

The Graduate School

EASTERN ILLINOIS UNIVERSITY

Thesis Maintenance and Reproduction Certificate

FOR: Graduate Candidates Completing Theses in Partial Fulfillment of the Degree
Graduate Faculty Advisors Directing the Theses

RE: Preservation, Reproduction, and Distribution of Thesis Research

Preserving, reproducing, and distributing thesis research is an important part of Booth Library's responsibility to provide access to scholarship. In order to further this goal, Booth Library makes all graduate theses completed as part of a degree program at Eastern Illinois University available for personal study, research, and other not-for-profit educational purposes. Under 17 U.S.C. § 108, the library may reproduce and distribute a copy without infringing on copyright; however, professional courtesy dictates that permission be requested from the author before doing so.

Your signatures affirm the following:

- The graduate candidate is the author of this thesis.
- The graduate candidate retains the copyright and intellectual property rights associated with the original research, creative activity, and intellectual or artistic content of the thesis.
- The graduate candidate certifies her/his compliance with federal copyright law (Title 17 of the U. S. Code) and her/his right to authorize reproduction and distribution of all copyrighted materials included in this thesis.
- The graduate candidate in consultation with the faculty advisor grants Booth Library the non-exclusive, perpetual right to make copies of the thesis freely and publicly available without restriction, by means of any current or successive technology, including by not limited to photocopying, microfilm, digitization, or internet.
- The graduate candidate acknowledges that by depositing her/his thesis with Booth Library, her/his work is available for viewing by the public and may be borrowed through the library's circulation and interlibrary loan departments, or accessed electronically.
- The graduate candidate waives the confidentiality provisions of the Family Educational Rights and Privacy Act (FERPA) (20 U. S. C. § 1232g; 34 CFR Part 99) with respect to the contents of the thesis and with respect to information concerning authorship of the thesis, including name and status as a student at Eastern Illinois University.

I have conferred with my graduate faculty advisor. My signature below indicates that I have read and agree with the above statements, and hereby give my permission to allow Booth Library to reproduce and distribute my thesis. My adviser's signature indicates concurrence to reproduce and distribute the thesis.

Graduate Candidate Signature

Fatemeh Khastkhodaardekani
Printed Name

Computer Technology
Graduate Degree Program

Faculty Adviser Signature

David Wayne Melton
Printed Name

12/07/15
Date

Please submit in duplicate.

Copyright @ 2015

Fatemeh Khastkhodaardekani

All rights reserved

Abstract

This thesis addresses a need for an online collaboration environment for the researchers at Eastern Illinois University (EIU). As the solution, it provides a portal for researchers (students and faculties) who wish to explore and collaborate on projects based on ideas from those who may not have the necessary skills. This is considered to help developing projects based on the ideas as well as making teams, and teach different groups of people different perspectives. Therefore, it offers a learning environment. I started off by examining existing tools and methodologies to satisfy this purpose. As the result, the deliverable project is a portal implemented in PHP programming using XAMPP Server v3.2.1 with Oracle Enterprise Edition 11g R2 as the portal's database. The main part of this thesis is working on the portal's designing phase in an optimized way to run a profile-based collaboration environment with PHP and Oracle.

Keywords: The PHP/Oracle project, Project Portal, Collaboration Portal

I dedicate this to my mom and my dad for appreciation of their love and supports.

تقدیم به پدر و مادر عزیزم به پاس عشق و تلاششان.

My sincere thanks go to my thesis advisor, Dr. Israr.

Besides him, I would like to thank my thesis committees: Dr. Liu and Dr. Wahby, for their leadership and teaching.

Table of Contents

ABSTRACT..... III

LIST OF FIGURES..... VIII

LIST OF TABLES X

CHAPTER 1: INTRODUCTION1

 1.1 STATUS OF THE PROBLEM..... 1

 1.2 DEFINITION OF TERMS 2

 1.3 SCOPE OF THE THESIS RESEARCH..... 3

 1.4 THESIS CONTRIBUTION 3

 1.5 THESIS CONTENT..... 4

CHAPTER 2: LITERATURE REVIEW5

 2.1 CURRENT TECHNOLOGY 5

 2.2 PROGRAMMING LANGUAGES 6

 2.3 DATABASE 8

CHAPTER 3: COLLABORATION PORTAL.....10

 3.1 DESIGNING PHASES 25

 3.2 FRONTEND DESIGN..... 25

 3.3 BACKEND STORAGE DESIGN 30

 3.3.1 *Resource Tables*..... 31

 3.3.2 *Other Tables* 33

 3.3.3 *Sample Data* 42

CHAPTER 4: IMPLEMENTATION PHASE57

 4.1 INSTALLING ORACLE DATABASE AND XAMPP SERVER..... 57

 4.2 CONNECTING XAMPP SERVER AND ORACLE DATABASE 58

 4.3 CREATING PHP TEMPLATE..... 61

 4.4 CREATING AND POPULATING TABLES..... 72

 4.5 ORACLE TABLE RELATIONSHIPS..... 74

 4.6 PHP/ORACLE CODING 76

CHAPTER 5: TESTING PHASE77

CHAPTER 6: CONCLUSION.....80

 6.1 FUTURE WORK 80

REFERENCES82

APPENDIX A: CSS AND JS CODES85

 A-THE CSS CODE 85

 A-JS CODES 100

APPENDIX B: PHP TEMPLATE CODES103

 B-FUNCTION.PHP.....103

 B-HEADER.PHP.....104

 B-FOOTER.PHP106

 B-SIDEBAR.PHP107

 B-CLASS-THEM-METHOD.PHP107

APPENDIX C: PHP/ORACLE FUNCTIONS.....109

APPENDIX D: PHP/ORACLE FILES.....121

 D-HOME.PHP.....121

 D-PROFILE.PHP123

 D-PROJECT.PHP.....125

 D-IDEA.PHP127

 D-CIRCLE.PHP128

 D-ADMIN.PHP.....131

 D-SEARCH.PHP.....132

 D-LOGOUT.PHP.....133

APPENDIX E: SQL/PLUS QUERIES134

 E-CREATING TABLES AND RELATIONSHIPS134

 E-TABLE POPULATING158

List of Figures

| | |
|--|----|
| Figure 1 – User access after logging in to home | 11 |
| Figure 2 - User access after logging in to profile | 12 |
| Figure 3 - User access after logging in to ideas | 13 |
| Figure 4 - User access after logging in to projects | 14 |
| Figure 5 - User access after logging in to circles | 15 |
| Figure 6 - System Level Flowchart | 16 |
| Figure 7 - Login level flowchart..... | 18 |
| Figure 8- Forgot your password | 19 |
| Figure 9 – Sign up level flowchart | 20 |
| Figure 10 - Home access level | 21 |
| Figure 11 – Profile access level..... | 22 |
| Figure 12 - Idea access level | 23 |
| Figure 13 – Project access level..... | 23 |
| Figure 14 – People (circle) access level | 24 |
| Figure 15 – Project creation action | 24 |
| Figure 16 – Idea creation..... | 25 |
| Figure 17 – The homepage initial drawing..... | 26 |
| Figure 18 – Drawing of the first page after user logs in | 27 |
| Figure 19 - The user’s profile drawing..... | 27 |
| Figure 20 – The user’s project page initial drawing | 28 |
| Figure 21 – The user’s ideas page initial drawing..... | 28 |

| | |
|---|----|
| Figure 22 – The initial drawing of users’ circles | 29 |
| Figure 23 – The initial Photoshop designing for graphic part | 30 |
| Figure 24 – General code for PHP/Oracle connection | 60 |
| Figure 25 – Specific coding for PHP/Oracle connection..... | 61 |
| Figure 26 – The index page of the website | 62 |
| Figure 27 – Home page after login | 63 |
| Figure 28 – CSS and JS files in the website root | 64 |
| Figure 29 – A part of CSS code | 65 |
| Figure 30 – The whole used JavaScript code | 66 |
| Figure 31 – use of HTML5 email validation..... | 67 |
| Figure 32 – The template files which are read by pages | 68 |
| Figure 33 – The pages which are reading the template files..... | 68 |
| Figure 34 – The header.php codes | 69 |
| Figure 35 – Circle.php codes..... | 71 |
| Figure 36 – Resource tables’ codes | 72 |
| Figure 37 – Other SQL tables rather than resource tables | 73 |
| Figure 38 – Alteration code for creating relationship between DOB_TBL and USER_TBL..... | 74 |
| Figure 39 – Work table alteration by adding foreign keys..... | 75 |
| Figure 40 – Relational table for USER_TBL and ROLE_TBL..... | 76 |
| Figure 41 – Admin.php is read from the header.php since the user is admin..... | 77 |
| Figure 42 – My project page..... | 78 |
| Figure 43 – Circle page screen shot..... | 79 |

List of Tables

| | |
|--|----|
| Table 1 – Privacy table | 31 |
| Table 2 – Interest area table | 31 |
| Table 3 – Major table | 32 |
| Table 4 – Degree table | 32 |
| Table 5 – Skill table..... | 32 |
| Table 6 – Role table | 32 |
| Table 7 – Feedback table | 33 |
| Table 8 – User table | 33 |
| Table 9 – DOB (Date of Birth) table..... | 34 |
| Table 10 – User-role relation table..... | 34 |
| Table 11 – User location table | 34 |
| Table 12 – Work table | 35 |
| Table 13 – User-interest relationship table..... | 35 |
| Table 14 – Phone table | 36 |
| Table 15 – Link table | 36 |
| Table 16 – Education table..... | 37 |
| Table 17 – User-skill relational table | 37 |
| Table 18 – Following table..... | 37 |
| Table 19 – Endorse table..... | 38 |
| Table 20 – Project table..... | 38 |
| Table 21 – User-project relation table | 39 |
| Table 22 – Project-skill relation table..... | 39 |
| Table 23 – Project-interest relation table..... | 39 |

| | |
|---|----|
| Table 24 – Project steps table | 40 |
| Table 25 – Step-user relation table | 40 |
| Table 26 – Relational table for users, feedback, and project..... | 40 |
| Table 27 – Idea table | 41 |
| Table 28 – Idea-project table | 41 |
| Table 29 – Idea-skill relation table..... | 41 |
| Table 30 – Idea-interest relation table..... | 42 |
| Table 31 – Comment table | 42 |
| Table 32 - Privacy sample table..... | 43 |
| Table 33 – Interest sample table | 44 |
| Table 34 – Major sample table..... | 46 |
| Table 35 – Degree sample table..... | 46 |
| Table 36 – Skill sample table | 47 |
| Table 37 – Role sample table | 47 |
| Table 38 – Feedback sample table..... | 47 |
| Table 39 – User table samples | 48 |
| Table 40 – User/role relationship table..... | 48 |
| Table 41 – Users’ date of birth..... | 49 |
| Table 42 – Users’ locations | 49 |
| Table 43 – Users’ work experience | 49 |
| Table 44 – User and interest areas’ relationship table | 50 |
| Table 45 – Phone table examples | 50 |
| Table 46 – Link table samples | 50 |
| Table 47 – Users’ sample of educations..... | 51 |

Table 48 – User/skills relationship examples51

Table 49 – Following table with sample data52

Table 50 – Endorsement table with sample data52

Table 51 – Project table.....53

Table 52 – User/project relationship table with data samples.....53

Table 53 – Project/interest table with sample data53

Table 54 – The projects’ steps samples.....54

Table 55 – The relationship data sample for users and steps.....54

Table 56 – The relationship sample data for feedbacking users in projects55

Table 57 – Idea table sample55

Table 58 – Idea/project sample data55

Table 59 – Idea/skill relationship sample55

Table 60 – Idea/interest example56

Table 61 – The comment table example56

Chapter 1: Introduction

Along with technology advancement, one of the most important parts of a small or big company is their project management section because of the need for planning and management for implementing all the designed projects. The project management part can be constructed by human resources like a group of people or a person, or it can be designed by artificial intelligences like new project portals, and typically the combination of both is seen in most companies. From these options, project management along with information technology systems can be a strong part of any organization. First of all, information technology systems make every physical process less time-consuming. Accessibility is another positive advantage of this method. Second, it makes both sides, human resources and artificial intelligence, make less errors and optimize the project time and cost.

1.1 Status of the Problem

There is a research group at Eastern Illinois University (EIU) in Technology Department active in research and projects in renewable energy subjects. Started from 2010, the group has been named EIU Mentorship Research Program (EIUMRP). Having a one-hour meeting every Wednesday, the group is mentored by Dr. Liu and other guest speakers and professors. In this group, researchers (mostly from Sustainable Energy Program) get together at an agreed time to discuss their new ideas, find their favorite project, and start their work in an individual/team environment. However, due to the time constraints (this collaborations happens once a week, for one hour), not all ideas get discussed nor all researchers get the opportunity to participate. Furthermore, there are other researchers (students and faculty alike) from other departments who would be open to the idea of research collaboration. Some of them may have the skills, but do not know

where to start gaining experience. Some have creative ideas and thoughts, but do not have the necessary skills. Most of these students do not have any idea about the fact that they can use these great thoughts for starting a good project and as a result their ideas are practically never used and wasted.

To make this improvement, this thesis proposes a portal for communications about the projects in a less time-consuming method and more inclusive ways and subjects, which would be available 24/7. This will encourage students at EIU to get involved in the projects from the very basic level of involvement like reading and commenting on the ideas to the more sophisticated ones that make students get interested and find themselves able to do the projects, define project steps, and manage a project. This includes the activities that help them exchange their ideas, determine the steps that they need to take for projects, understand the requirements for doing research, and gain practical experiences in their interested area.

1.2 Definition of Terms

Portal: a complex Internet website providing access to services or other sites.

Profile-based website: a website that contains user profiles to associate data with a specific user or a customized desktop environment.

PHP: PHP or Hypertext Preprocessor is a widely-used open source general-purpose scripting language that is especially suited for web development and can be embedded into HTML.

Enterprise Oracle 11g R2: Oracle Enterprise Manager (OEM or EM) is a set of web-based database tools aimed at managing software and hardware produced by Oracle

Corporation as well as by some non-Oracle entities. 11 g is the version and R2 shows that it is the second Release.

Project Management: Project management is the application of processes, methods, knowledge, skills and experience to achieve the project objectives.

1.3 Scope of the Thesis Research

This project is focusing on the PHP and Oracle development for a collaboration website. Initially, it has five main tabs in the menu that make it easy to access different services for users. The one that is concentrated on in this thesis is the project tab. By implementing the project services, it will be easy to implement other tabs since most of them use the same PHP functions that are programmed in the project operations.

1.4 Thesis Contribution

According to Project Insight, every project is technically split into five main phases, but this thesis provides only four phases because the last phase, the maintenance phase, is not included here. Each of these phases has some sub-phases and contains an important part for optimizing project costs. Based on these, the four main phases are research, design, implementation, and testing.

- The research phase explores the best way of implementation based on the idea of the thesis.
- The design phase includes two main parts: front-end graphic design and back-end storage design.
- The implementation phase, which is very dependent on the design phase, explains how to implement the designed project.
- The testing has a case study which will be tested through the scope of thesis.

1.5 Thesis Content

The research phase, in Chapter 2, is a review of the possible technologies that can be used for creating the portal, and the reason why the used technologies have been chosen among others. Chapter 3 talks about planning and designing the system. As an online application, the designing phase should distribute the planning strategies to both graphic and database designing. Chapter 4 is about implementing the project based on what has been designed. In Chapter 5, a case study has been tested to show the way the website works. Chapter 6 states the conclusion and the possible ways that the project can be continued in the future.

Chapter 2: Literature Review

Before starting the project, the project's requirement ranged from the needed software to the way of using them should have been estimated. Assessing the required software and comparing the cons and pros of each, helped me to be aware of the expectations from such a collaborating environment.

2.1 Current Technology

Because the portal is about projects, estimation, scheduling, and team working, the portal is considered as a project management portal. According to one of the recent articles by Hendricks (2015), there are a few top online applications for project management software:

- Wrike: "Wrike is an excellent project management program that comes in both online and app form. If you only have a few users, the platform is free, and the cost for more users is not prohibitive. It allows you to track your campaign status, run workload reports, meet deadlines, and keep account of time and money spent" (Hendricks, 2015).
- Trello: "Trello is another fine online program and app that's free and easy to use. It uses a method known as Kanban, which enables users to move "cards" around, as a method of representing tasks within a project" (Hendricks, 2015).
- Producteev: "Producteev is absolutely free for all projects and users as long as you don't need to integrate with other platforms. That requirement will entail an upgrade to the pro account" (Hendricks, 2015).

There are several project portals that can be found online. The similar idea at a university scale which is interesting to look at is *Liquid Planer*. The feature of this website

that makes it similar to our project is that it is based on project management, and communications. The difference between these two is that my project tries to find people based on their skills, and interest areas; however, in *Liquid planner*, people find each other through email.

2.2 Programming Languages

According to Alex (2015), the first top five programming language in 2015 are listed as follows:

1. Java

It is considered as the first in the list and has the highest position with Android Operation System. This programming language is utilized for mobile and desktop applications, enterprise level purpose, and establishing Android apps on tablets and smartphones.

2. PHP

With the help of PHP, one can enlarge a web app quickly and effortlessly. It is the foundation of many Content Management Systems (CMSs).

3. JavaScript

JavaScript can generate communication for websites. It is utilized for construction superb user interface and interactive responsive web pages including animations, images, scripts, and objects.

4. Python

It is called all-in-one language because it has the ability to expand web apps, data analysis, and user interface with frameworks. Python is vastly used by huge companies for evaluate data sets.

5. Objective-C

If anyone interested in creating iOS, they are recommended to use Objective-C language.

After doing more research, working with open source programs was chosen because they are more popular in the technology world. Thus, the graphic side of the thesis is decided to be implemented in PHP.

PHP or Hypertext Processor is a free server-side programming language. It has been created not only for web development purposes, but also for general programming platforms. In an article by Martin (2015), it is stated that this language was created about a decade ago, in 2004, and has powered 200 million websites around the world. The very famous websites that are using PHP are Facebook, WordPress, and Digg.com.

“PHP is an interpreted script language which means that it is usually processed by an interpreter. For this reason, the language is most suitable for server-side programming that has server tasks being repeatedly performed when the website development process is on” (Martin, 2015).

PHP features are not limited to these points mentioned above. There are other facts about PHP that make it a powerful programming language:

- Being open source which, according to Wikipedia, means the original source code is made freely available and may be restructured and optimized
- PHP is compatible with Linux, Mac, and Windows
- It is very famous for social media and media websites
- It is very easy to embed PHP in HTML and vice versa

These reasons made PHP to be ranked fifth on Entrepreneurial Insights and sixth in Tiobe websites. It is also known as one of the top 10 programming languages in an article by Cass (2015).

2.3 Database

There are several database technologies that one can use as a portal storage. The first three databases with respective features are:

1. "Oracle:

- The most popular cull for consequential commercial projects and the oldest major database in the market (not compulsorily a disadvantage)
- Available in four different editions.
- More operating system flexibility than its Microsoft obverse
- The widest flexibility when it comes to the fortified server operating systems
- Virtual Private Database, Data Sentinel for standby database, automatic recollection storage and undo management

2. MySQL:

- Enterprises can commence out utilizing the free community server and later upgrade to the commercial version
- Runs on Linux, Windows, OSX and FreeBSD and Solaris
- Intuitive graphical utilizer interface for designing database tables
- Due to its open-source community, MySQL has a sizably voluminous bank of tutorials and information to avail you get commenced and solve quandaries
- Support for partitioning and replication, as well as for Xpath and for stored procedures, triggers and views

3. Microsoft SQL Server:

- The most widely used commercial DBMS
- Constrained to Windows, but this is an advantage if your enterprise uses mostly Microsoft products” (“Top 10 databases you should learn in 2015,” 2015).

To conclude, Oracle is chosen for this project because plus the mentioned features, it is the most popular official database in the United States of America. (“Oracle holds #1 DBMS market share worldwide for 2013”, 2014)

Chapter 3: Collaboration Portal

Deciding for using what technologies for a portal, based on the Literature Review, Chapter two, was the first step of the path. For this chapter, the portal is explained in more details with designing and planning.

The collaboration portal is a profile-based website. Users can have their profile, share any information of that with different privacies, like public, private, team members, and specific users in the system. Thus, they have a right menu in their account that helps them access “Home”, “Profile”, “My Projects”, “My Ideas”, and “Circles”.

“My projects” are simply the projects that user has created, or the projects that she/he is playing a part in them. For instance, if user 1 creates projects 1, there is a need to determine the teammates as well. Any chosen teammate will have project 1 in their profile just like user 1 does. “My ideas” has the same conception except the fact that ideas do not need any teammate. They can only be shared with different people or not even get published and be saved for user memos.

“Home” is a combination of “My projects” and “My ideas” which led the user to see all the changes and updates in them in her/his timeline.

“Circles” gives the ability to search and find other registered users according to the skills, interest areas, and even educations. In this way the user can get start communication and find relevant people with necessary skills. There is also a list of people in this menu who are either follower or followed users. The user can follow or unfollow users by accessing this menu.

The actions that can be done in the first page is login, sign up (if the user is not registered), take a look at an informative video, read the “About” page, and go through the website policies. Figures 1 to 5 show the big picture of the portal.

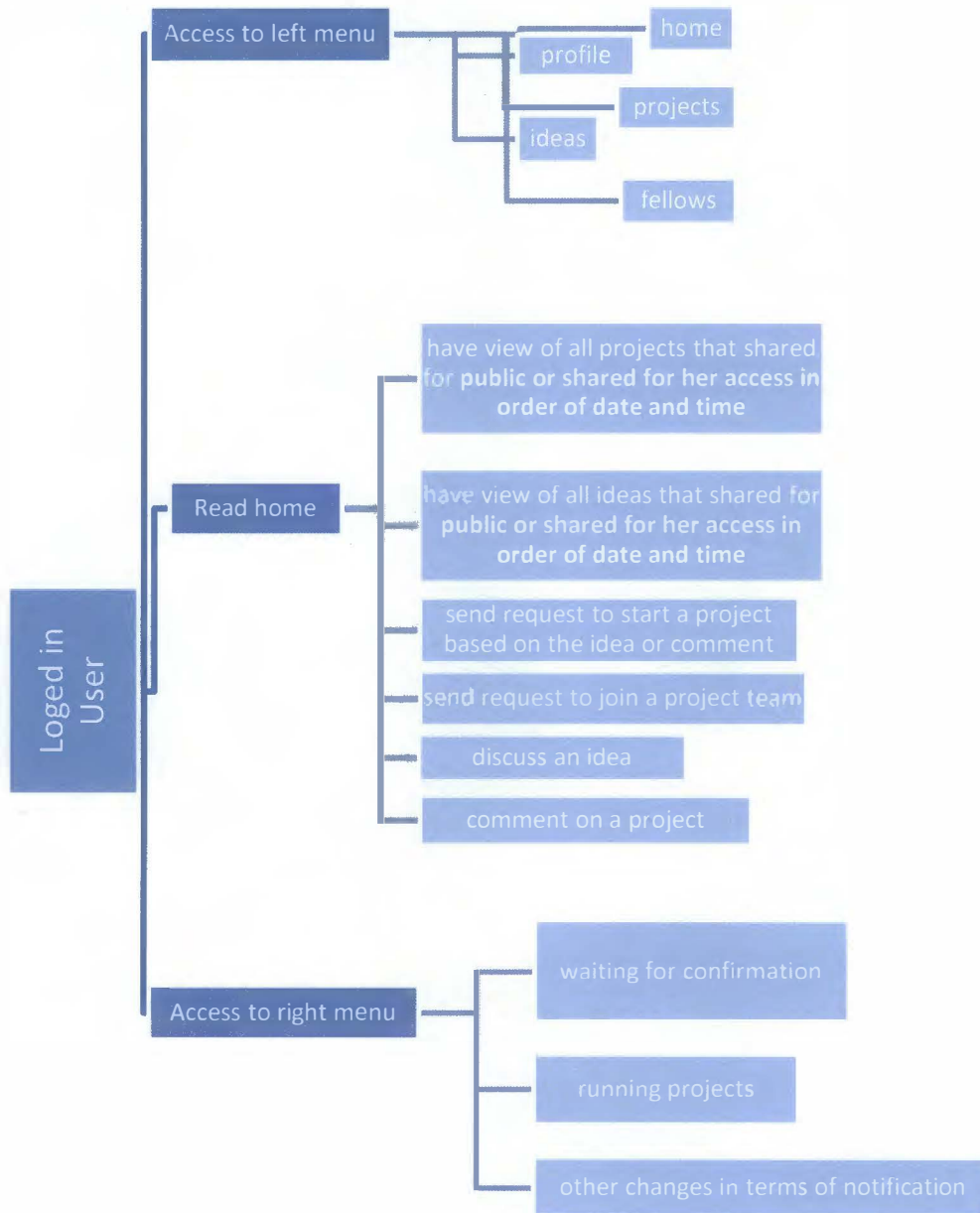


Figure 1 - User access after logging in to home

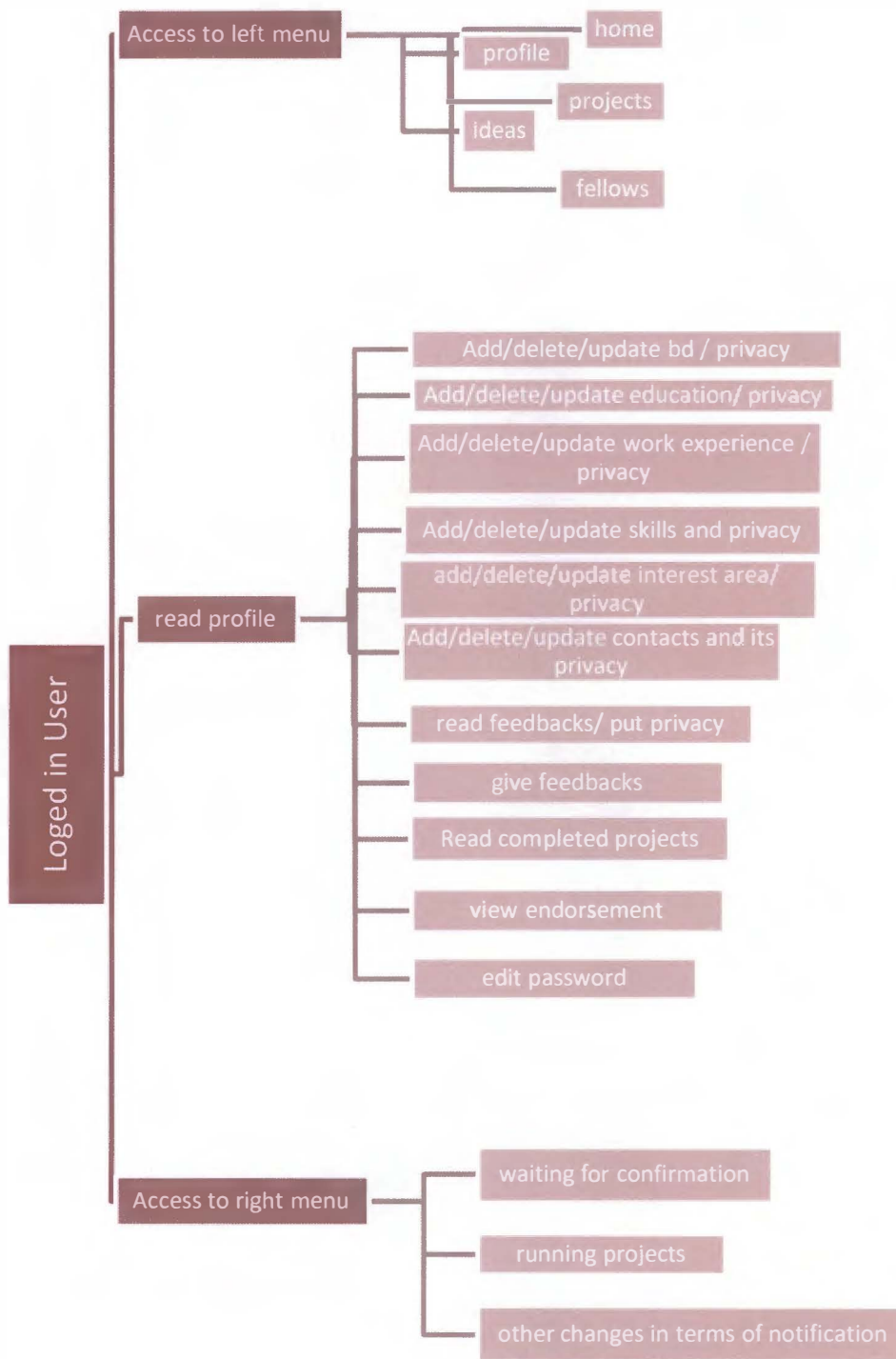


Figure 2 - User access after logging in to profile

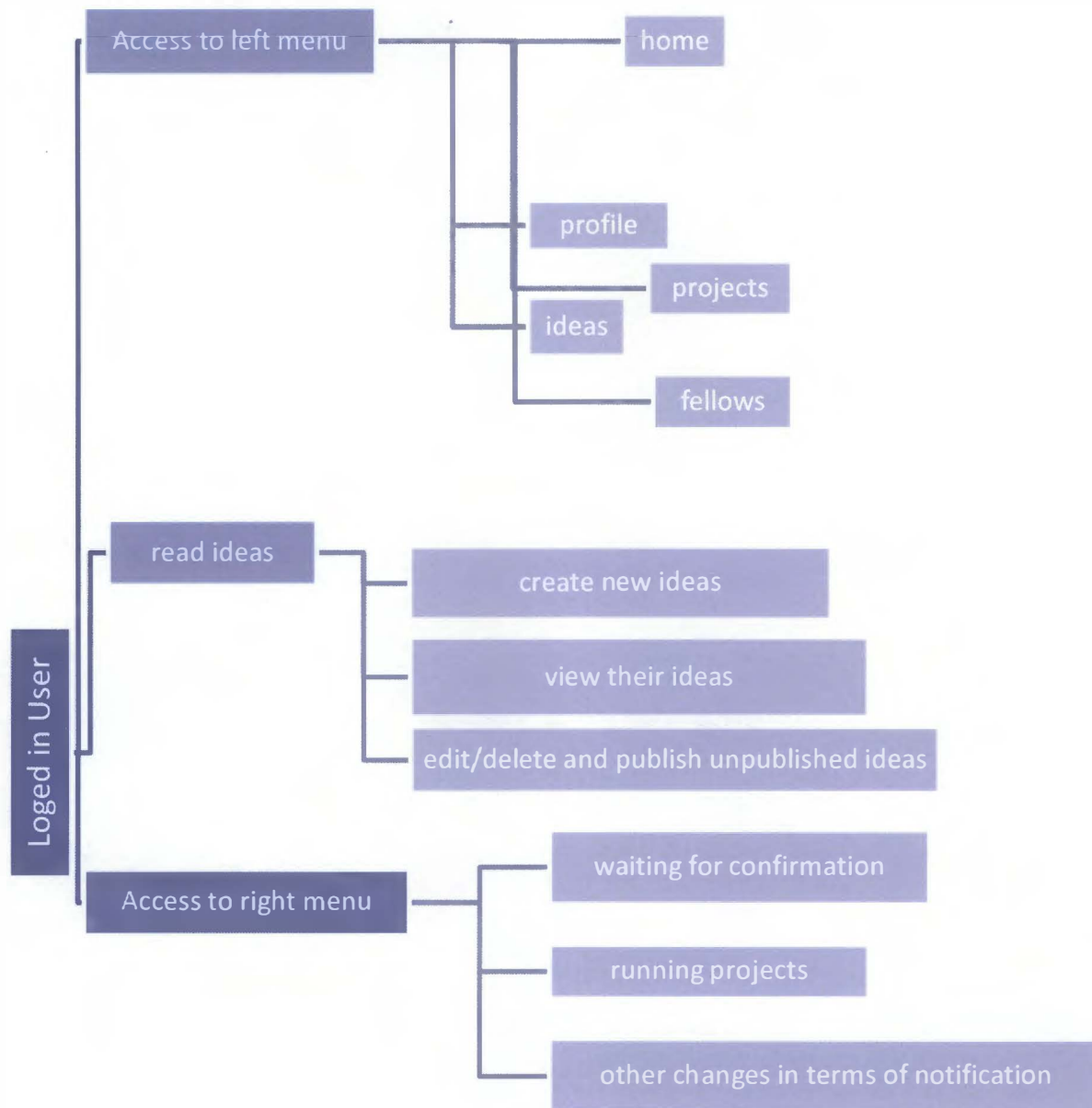


Figure 3 - User access after logging in to ideas

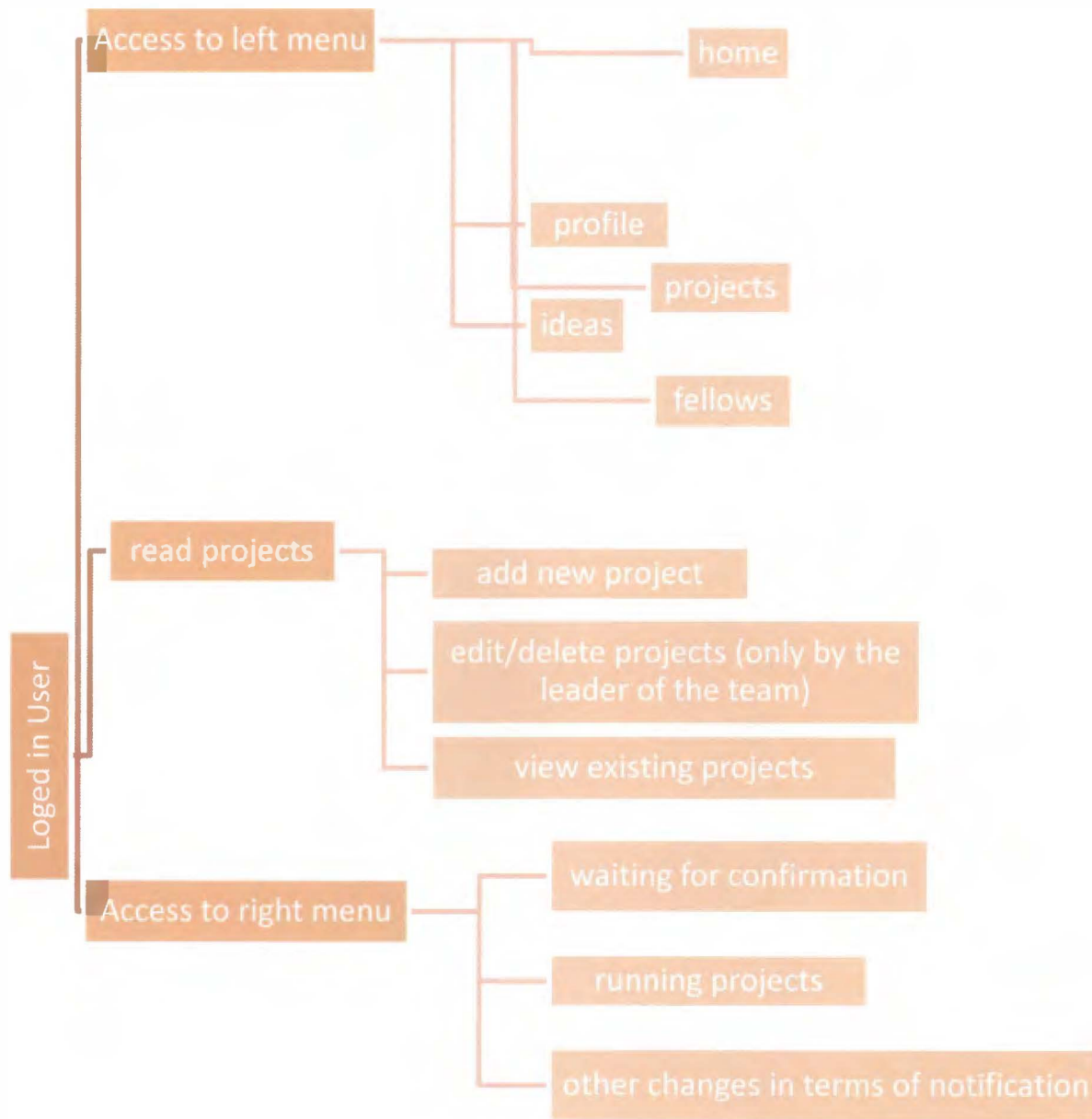


Figure 4 - User access after logging in to projects

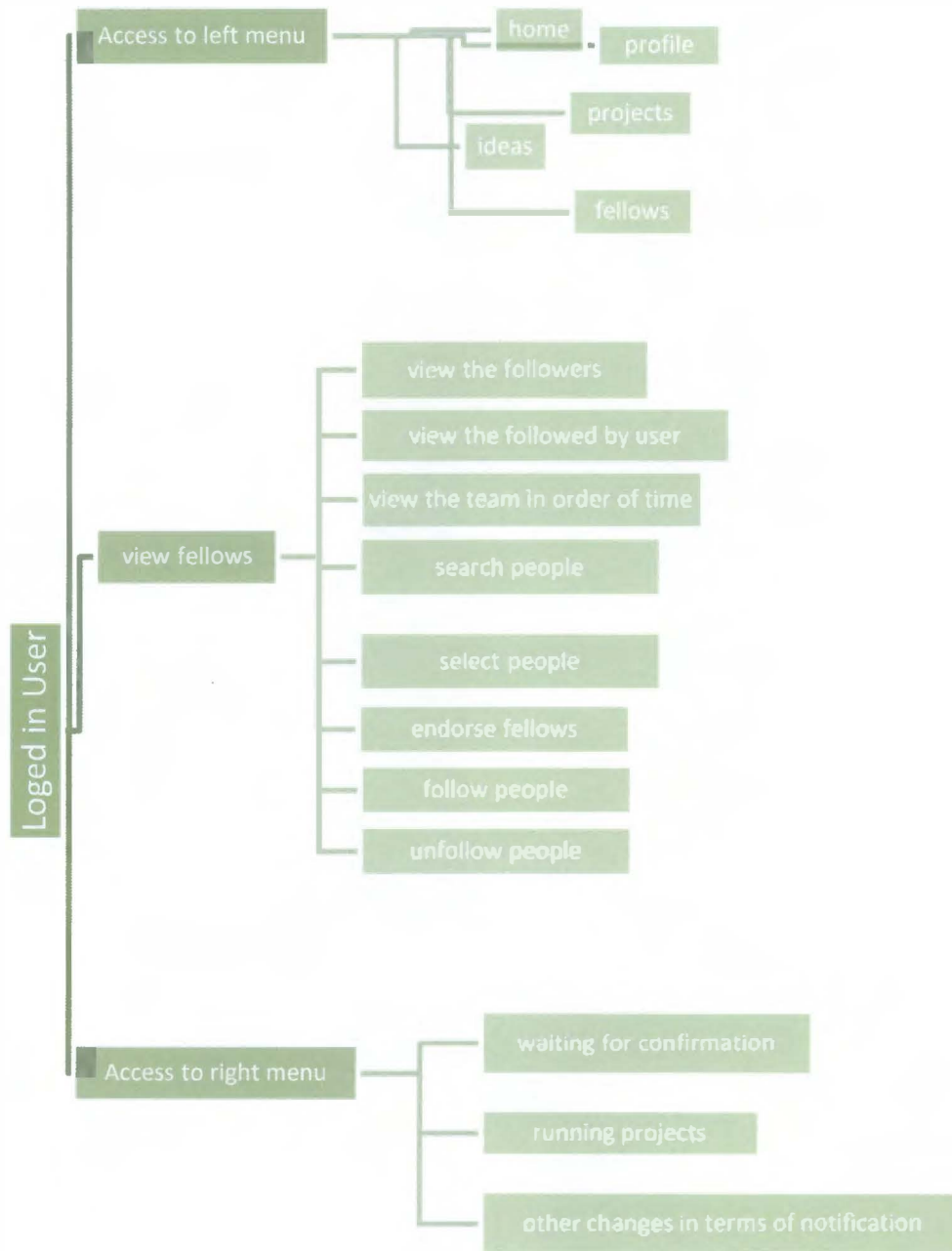


Figure 5 - User access after logging in to circles

Analyzing any action in the portal is very useful for designing phase. Based on Figures 1 to 5, the flowcharts can be created in different levels. The very general level is the system level. This level describes the actions in general regardless of details. Figure 6 is a flowchart in the system level.

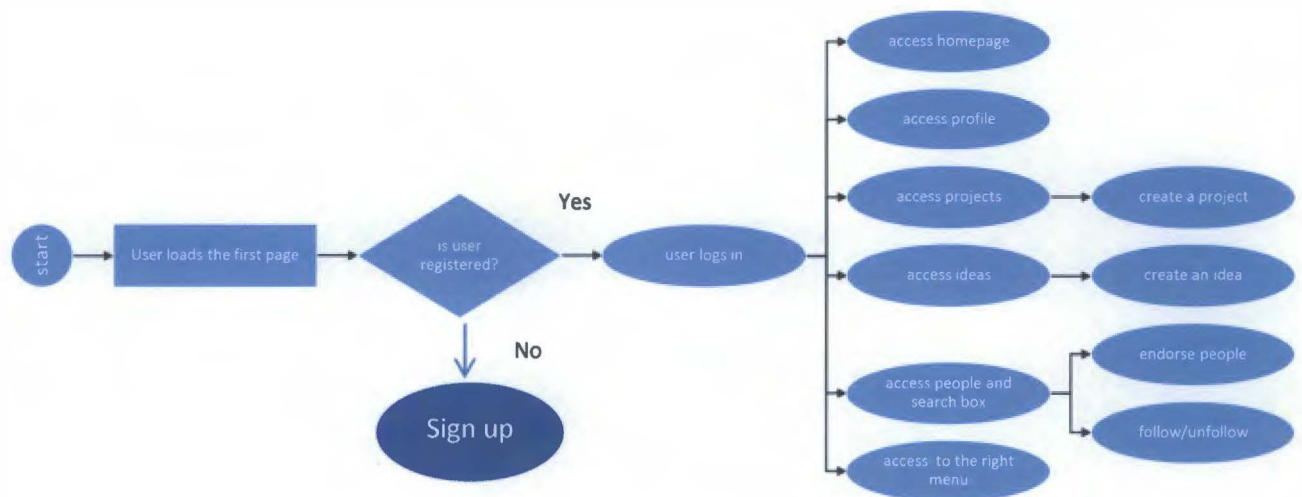


Figure 6 - System Level Flowchart

All of the actions, which are shown in the shape of ovals, can be decomposed and analyzed further. The following are analyzed in detail:

- Log in
- Sign up
- Forget the password
- Access to home
- Access to profile

- Access to ideas
- Access to projects
- Access to people and search box
- Access to the right side menu
- Create a project
- Create an idea
- Endorse people
- Follow/ unfollow

Figures from 7 to 16 are the figures of each action in the system. They show how users might act in different levels and how the roles permit each user to do those actions. The first one is login action flowchart. The details of login action is in Figure 7.

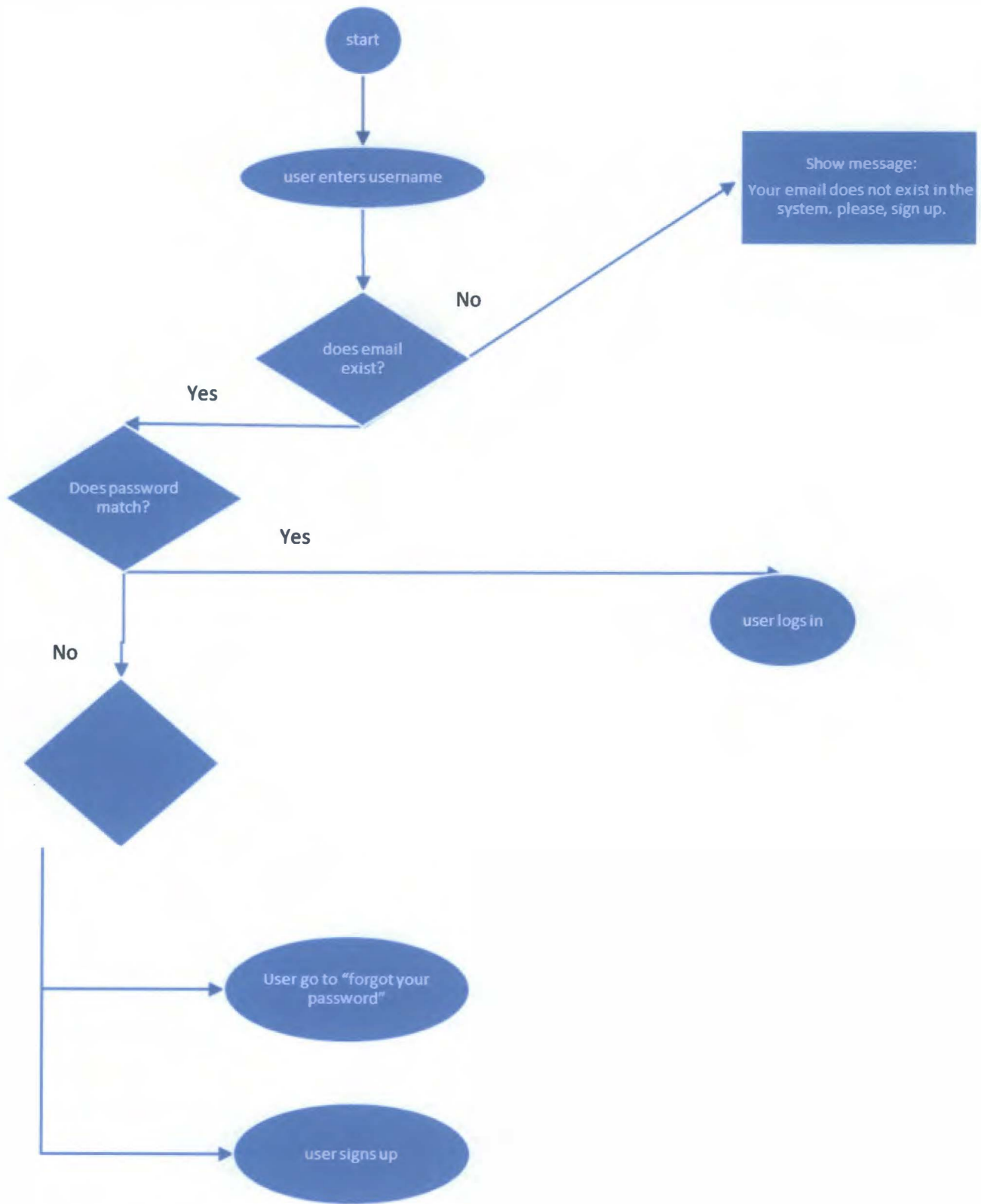


Figure 7 - Login level flowchart

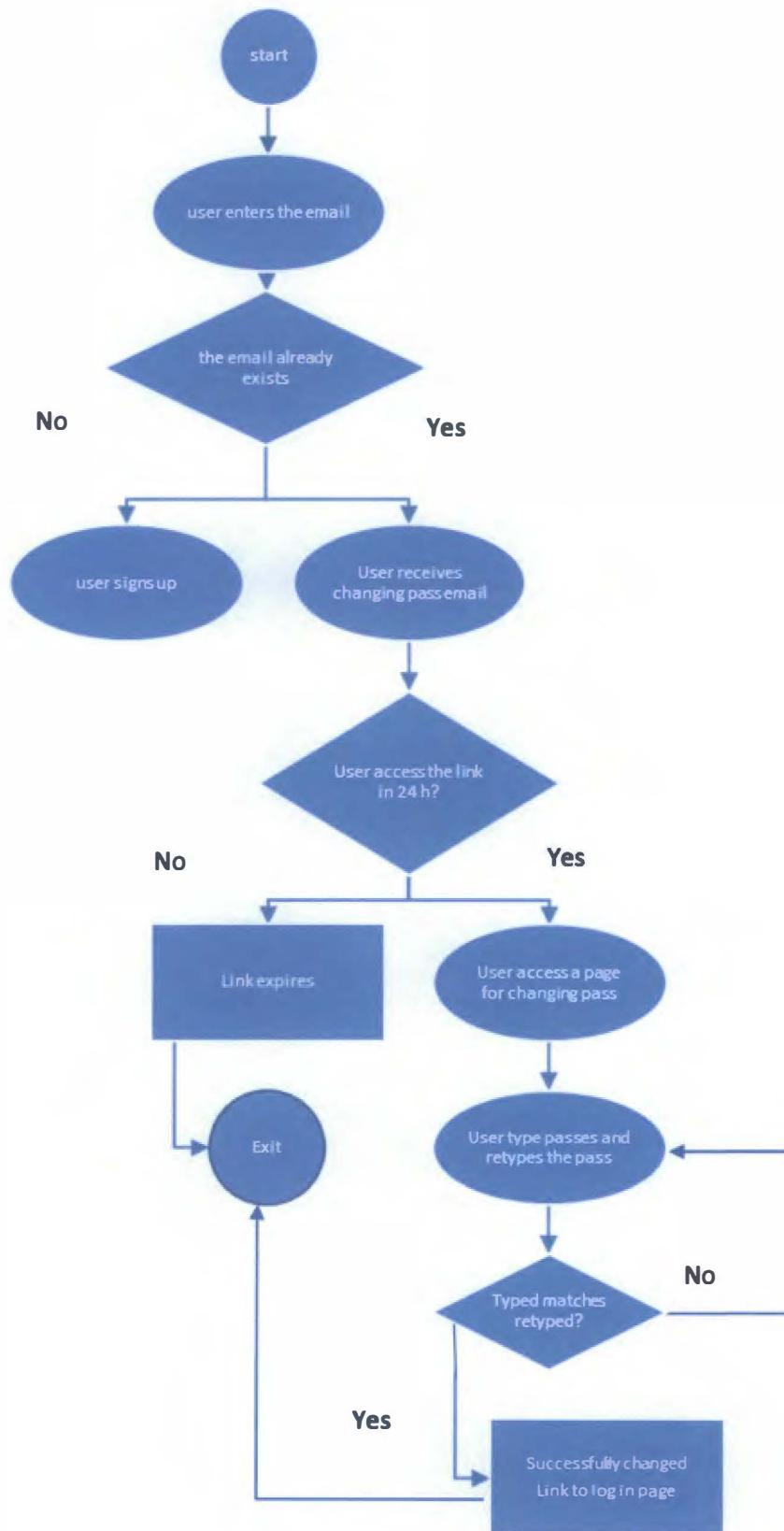


Figure 8- Forgot your password

Sign up level:

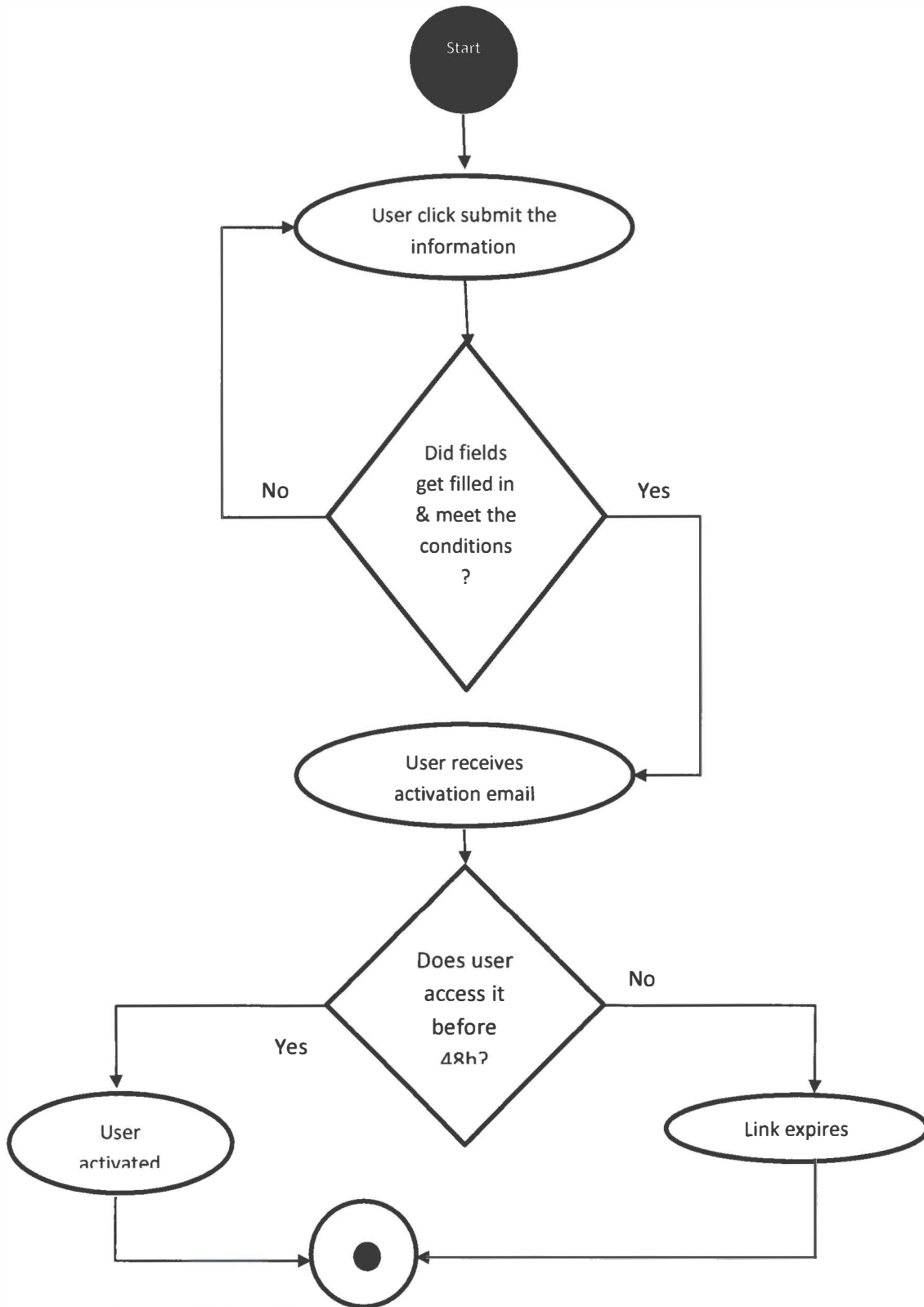


Figure 9 – Sign up level flowchart

Home Access Level:

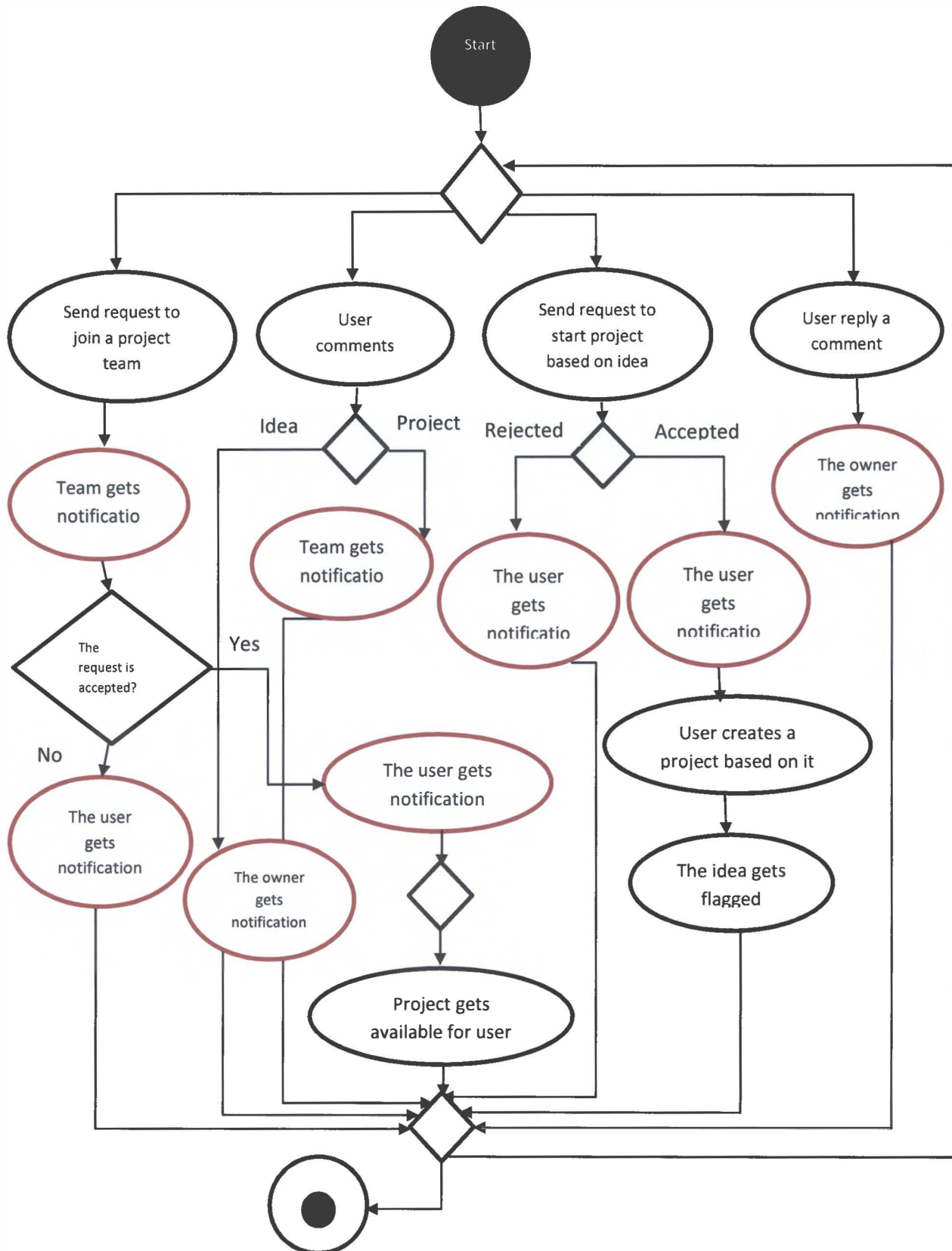


Figure 10 - Home access level

From the flowchart in Figure 10, project creation is an action that needs to be analyzed as well. Profile access is shown in Figure 11.

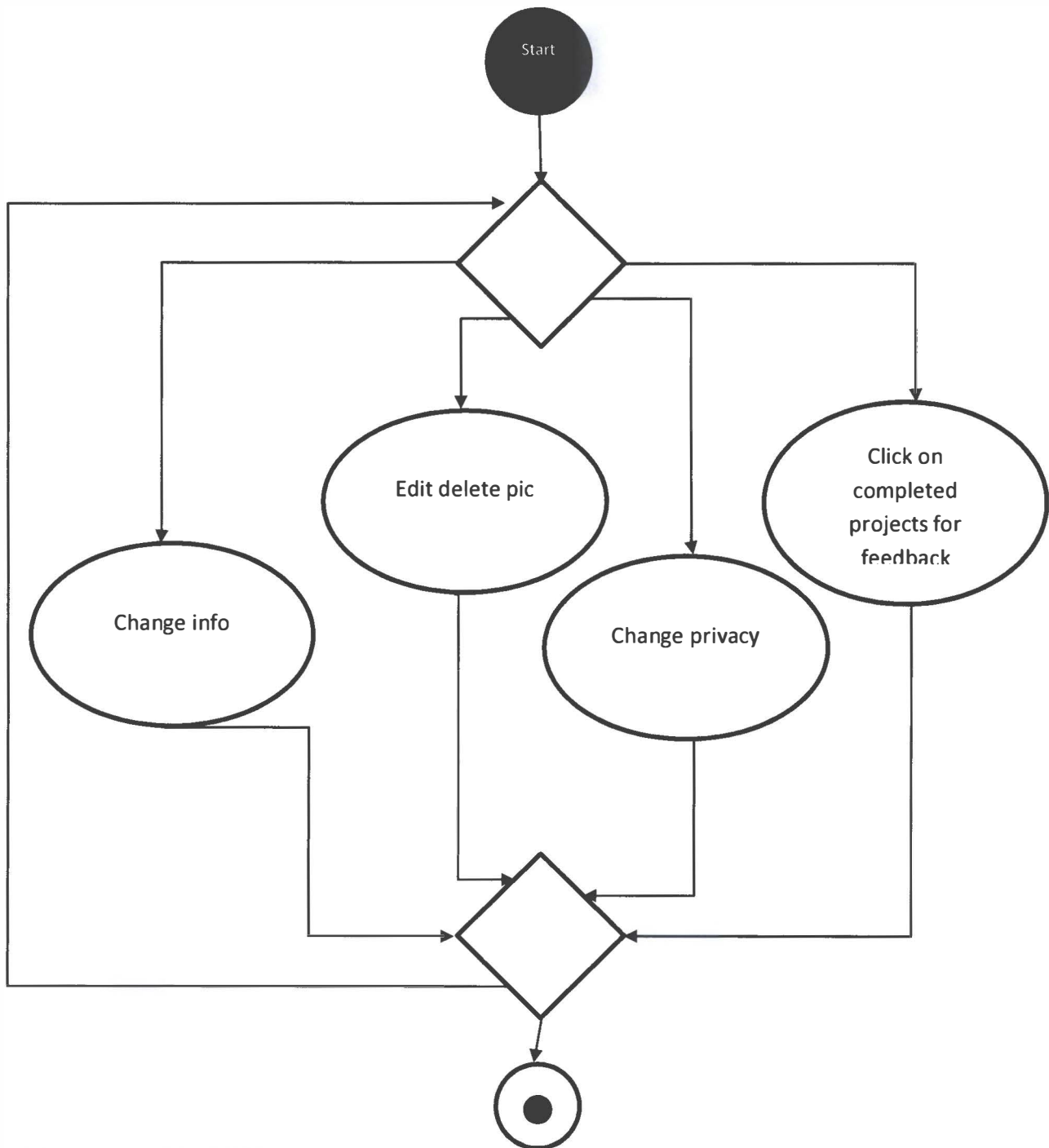


Figure 11 – Profile access level

Idea Access Level:

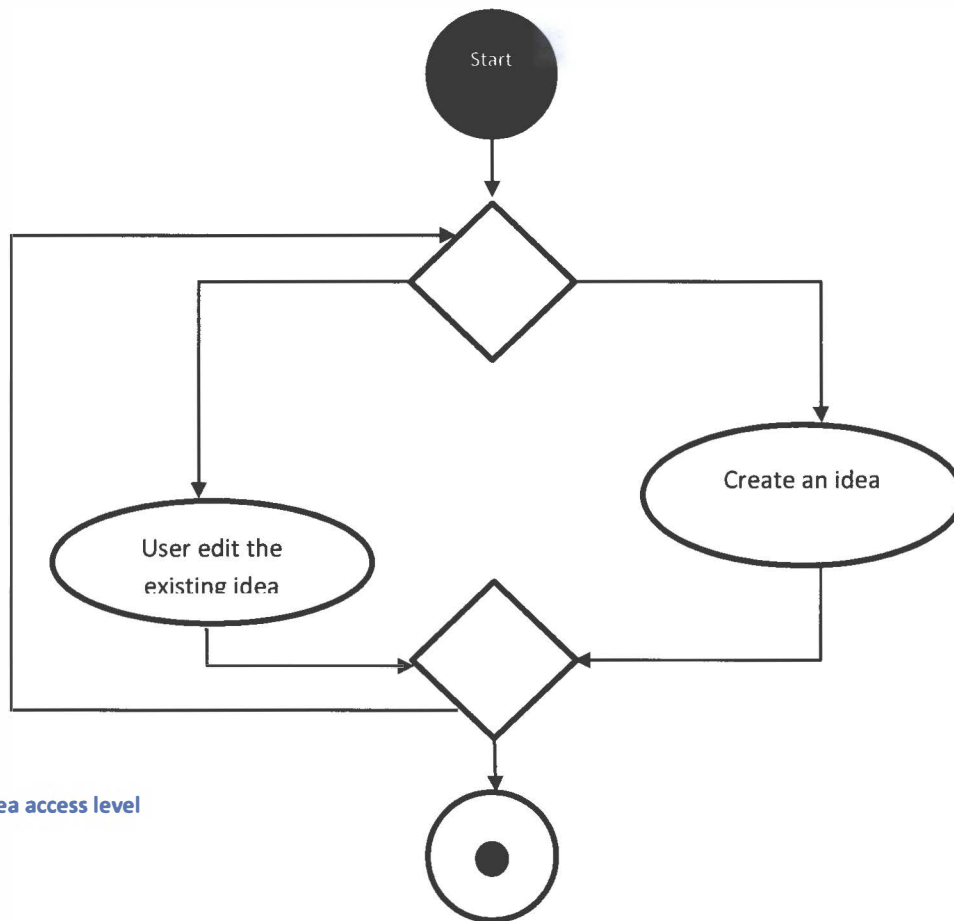


Figure 12 - Idea access level

Project Access Level:

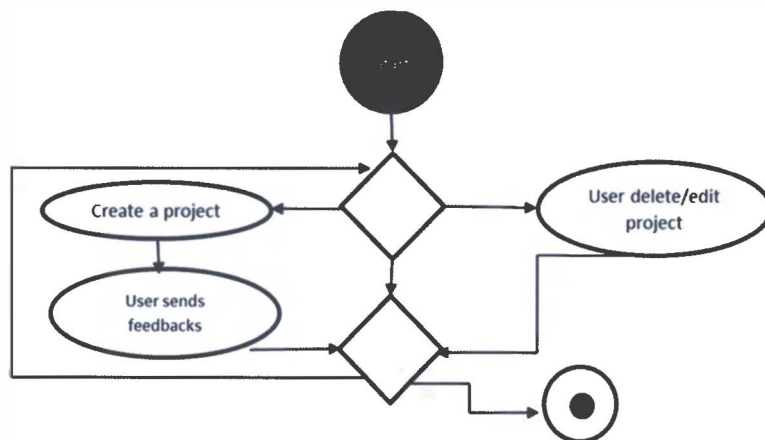


Figure 13 – Project access level

People Access Level:

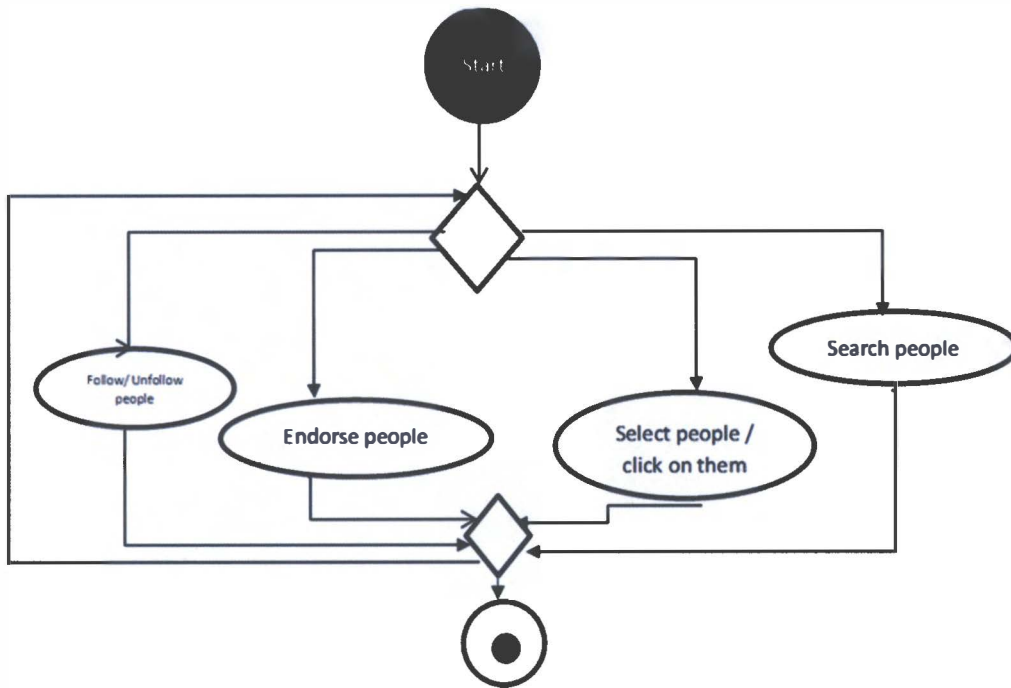


Figure 14 – People (circle) access level

Project Creation:

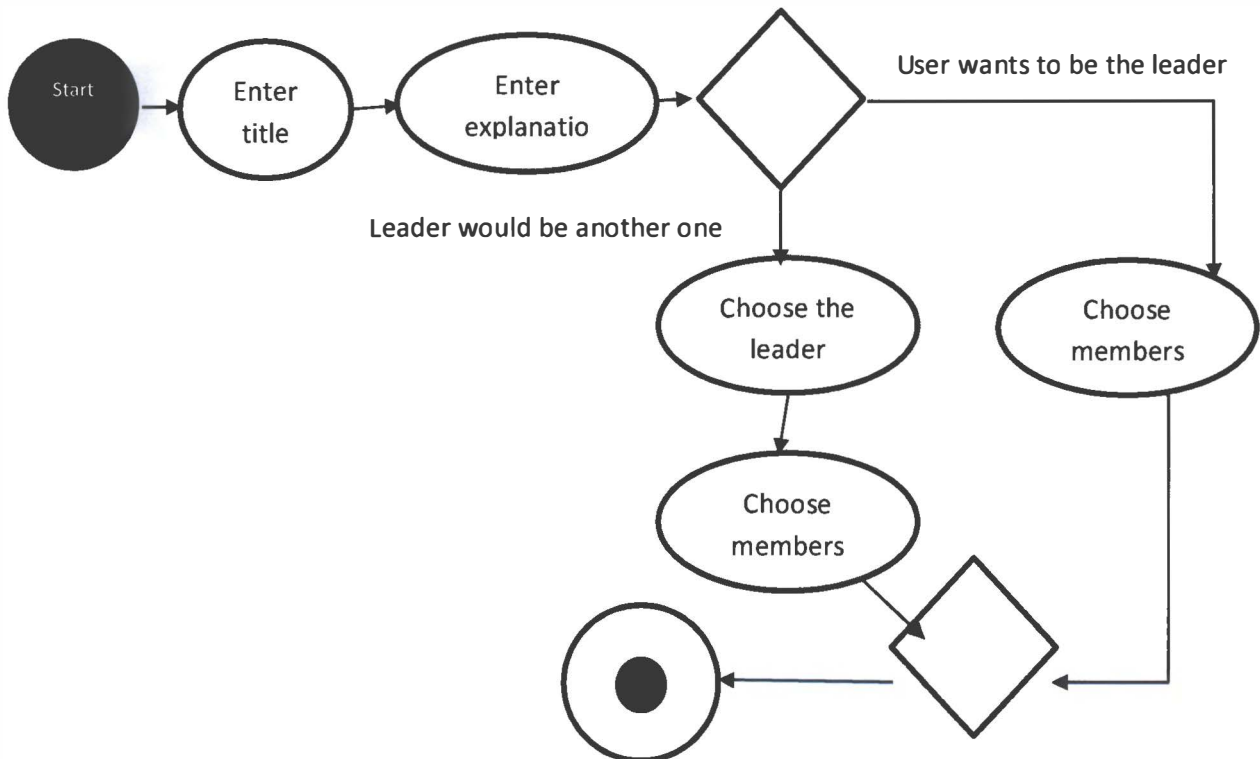


Figure 15 – Project creation action

Idea Creation:

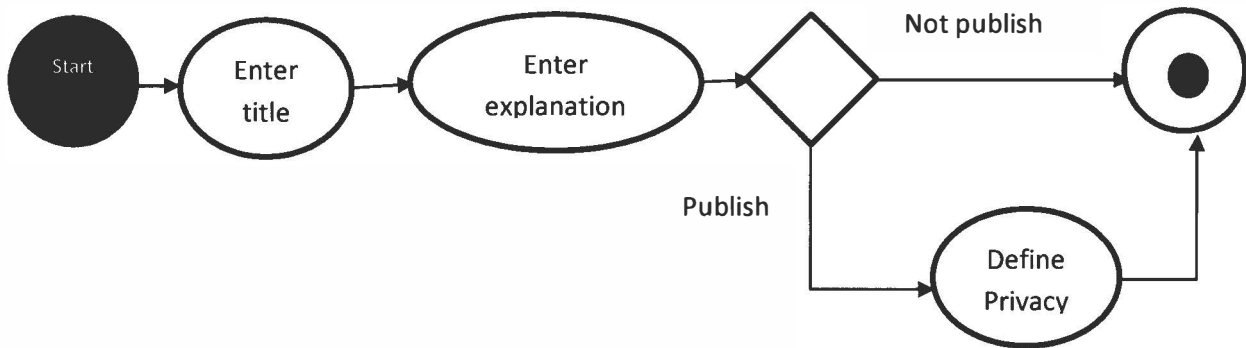


Figure 16 – Idea creation

After understanding the relationship and actions in more details, it is required to design a frontend and backend. Designing phase is the place that explains about them.

3.1 Designing Phases

For the explained portal, it is needed to plan ahead to prevent mistakes by having the road map for the implementation. Designing phase include two important steps.

1. Frontend Design
2. Backend Storage Design

3.2 Frontend Design

Creating the graphic design of the website is completely dependent on the original imagination of the application. Figures 17 to 22 show the pages from initial drawings.

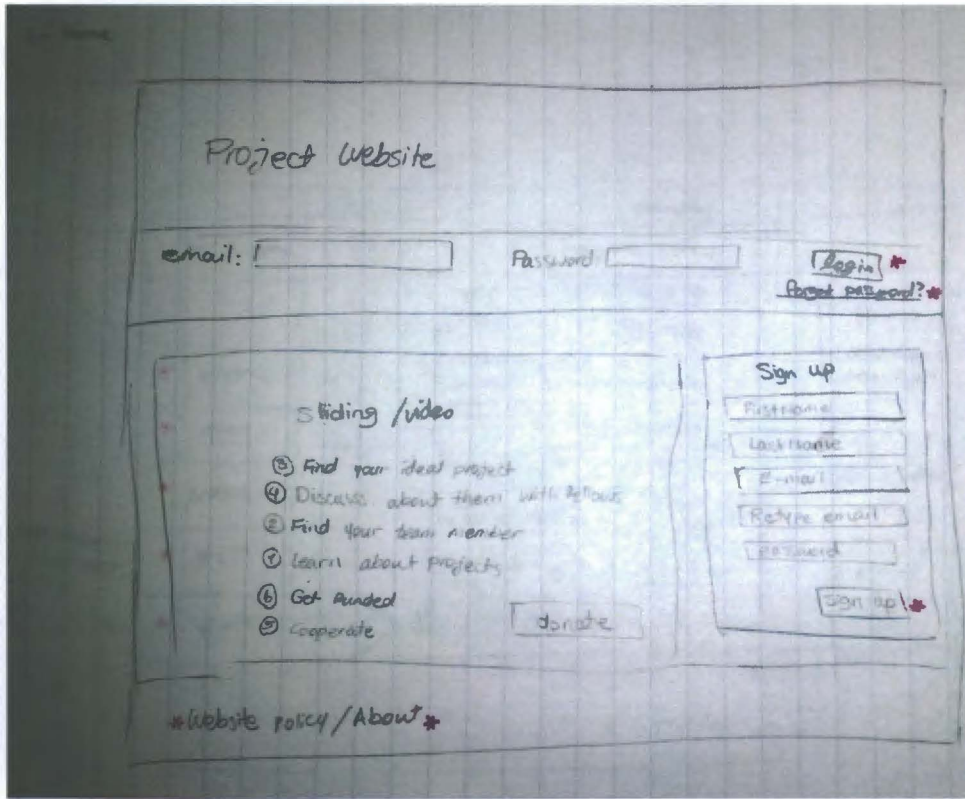


Figure 17 - The homepage initial drawing

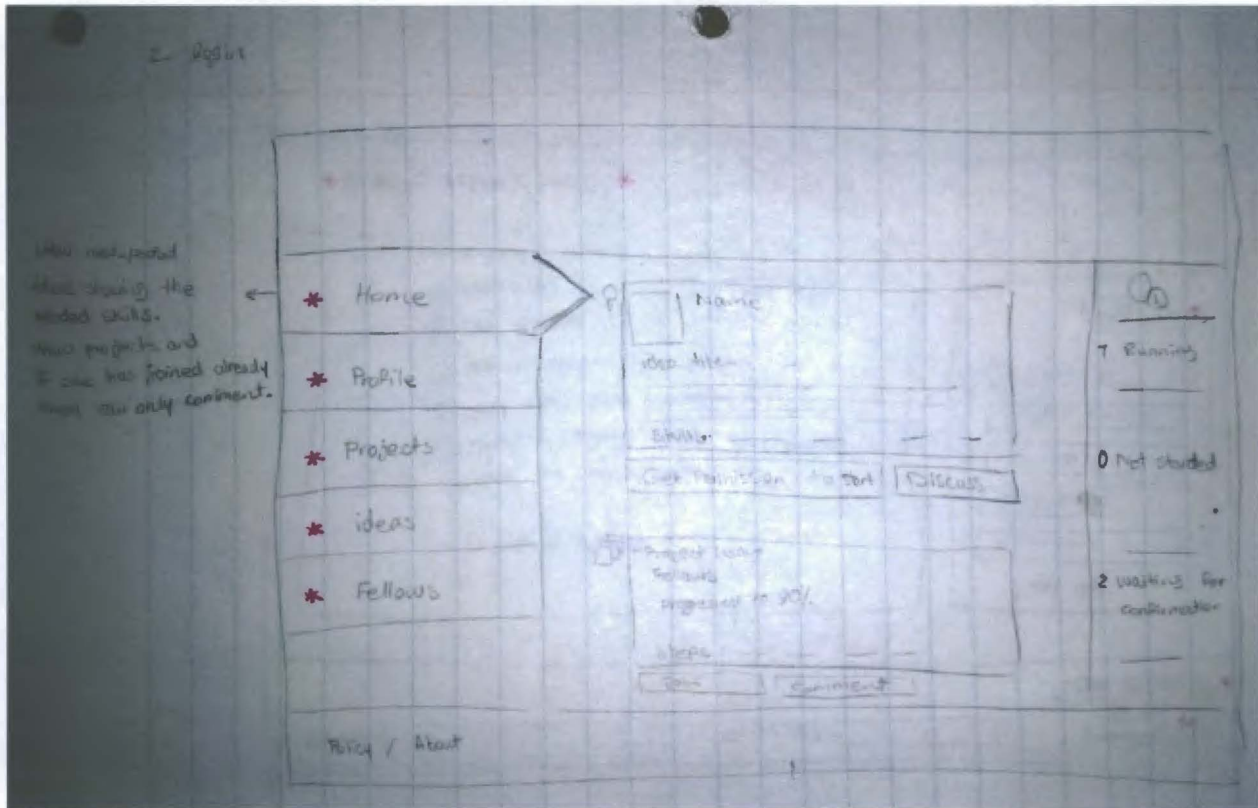


Figure 18 - Drawing of the first page after user logs in

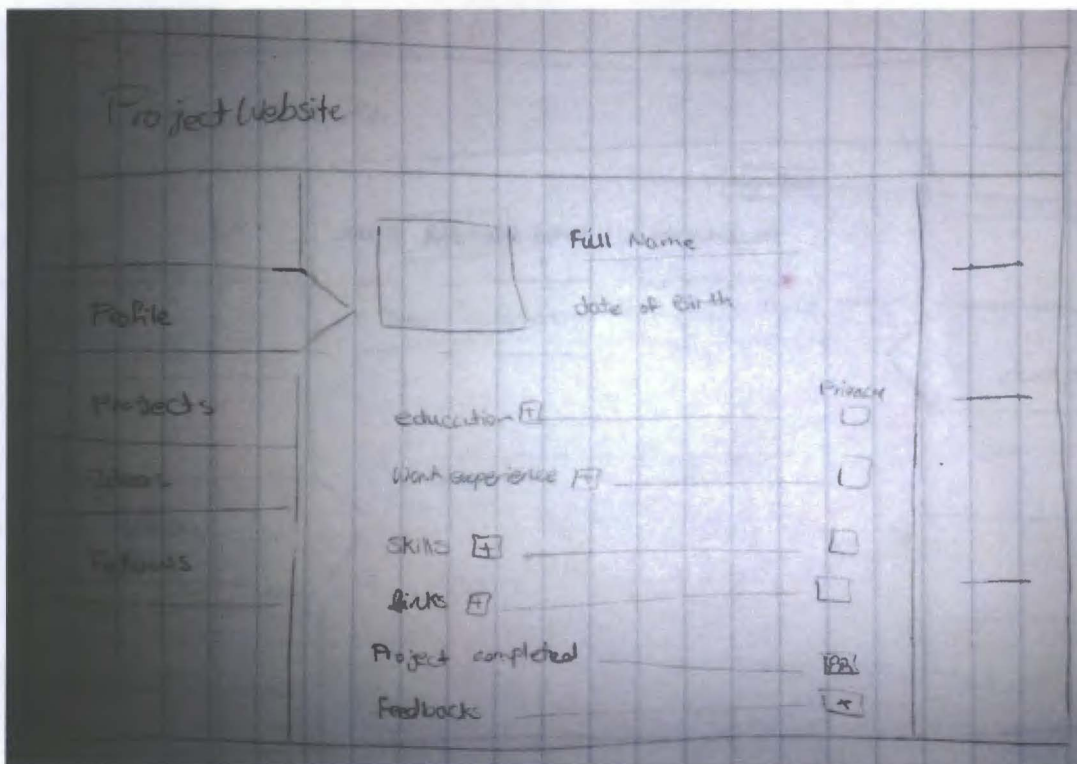


Figure 19 - The user's profile drawing

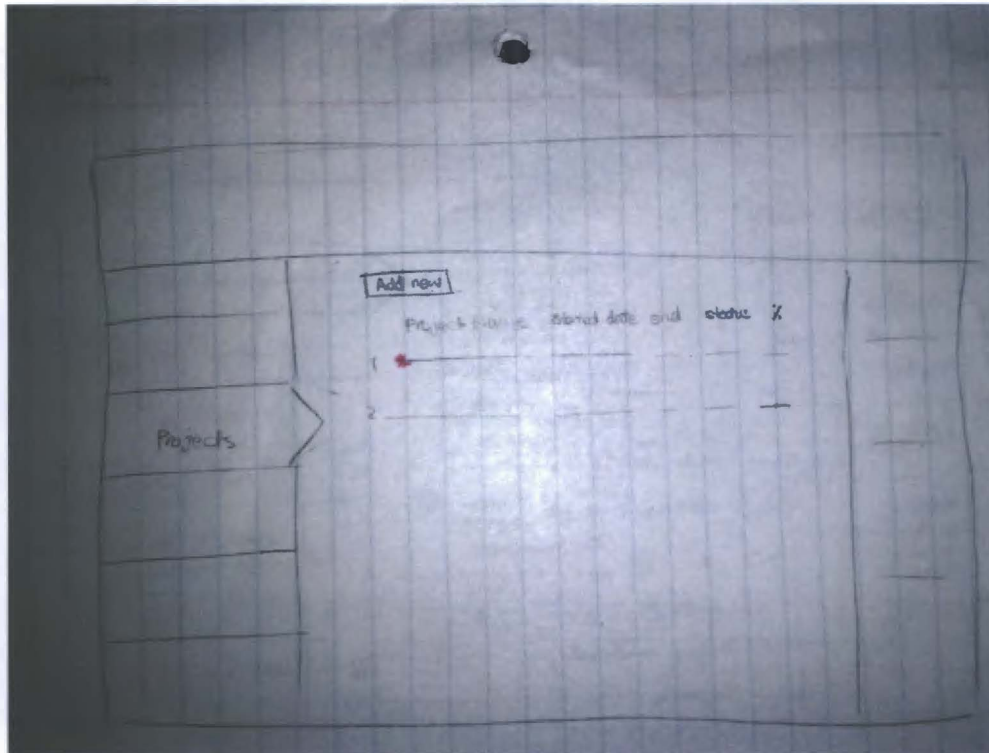


Figure 20 - The user's project page initial drawing

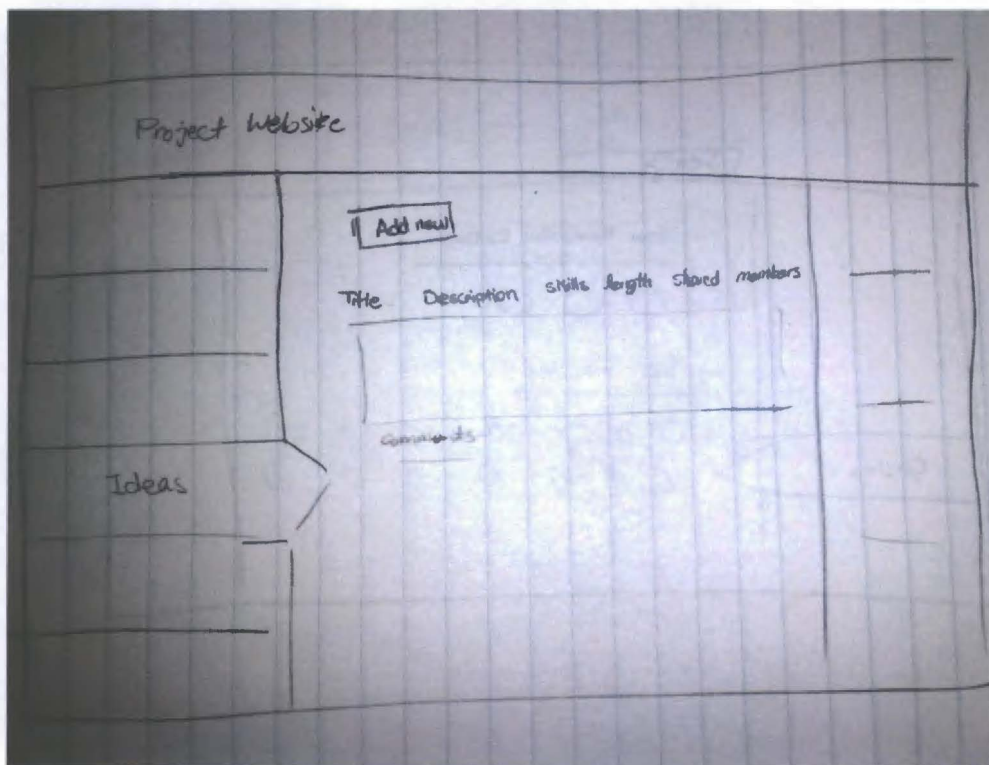


Figure 21 - The user's ideas page initial drawing

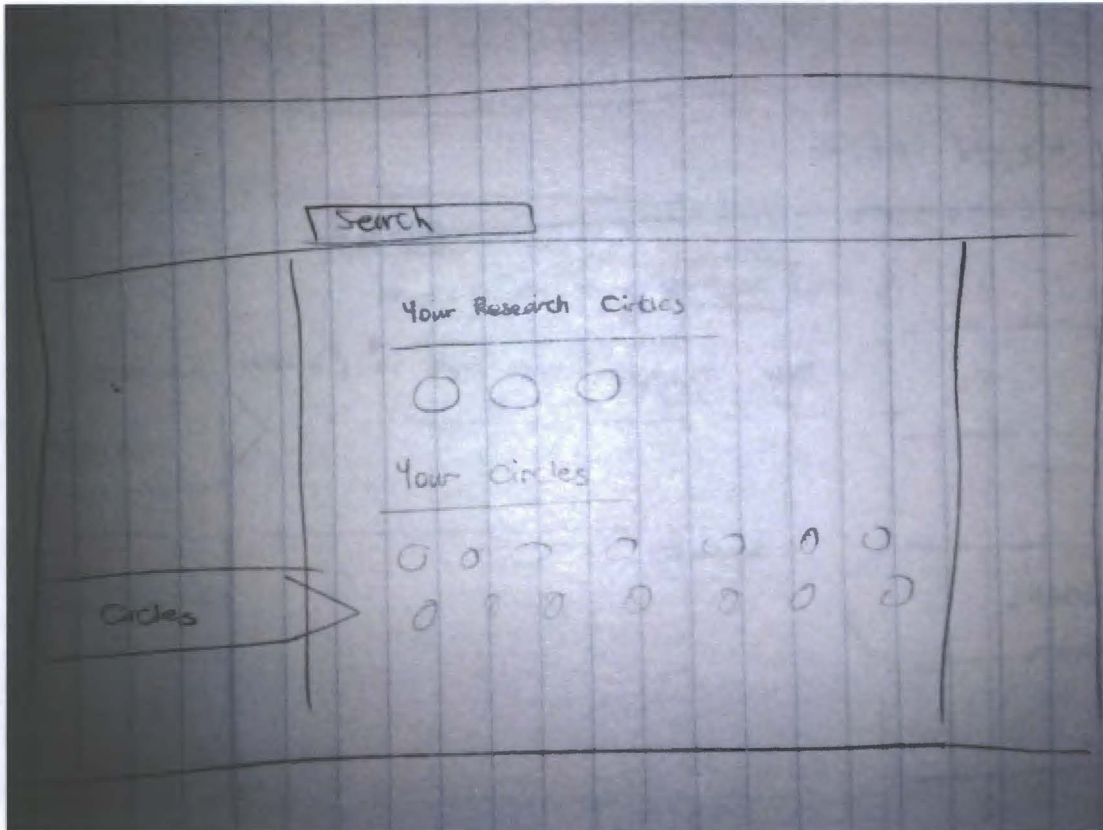


Figure 22 - The initial drawing of users' circles

Before starting to design the actual graphic, it would be helpful if the programmers take a look at some templates on the Internet. This gives them better ideas for the colors and shapes to start designing the initial graphics in Photoshop. My initial designing in Photoshop turned out to be like Figure 23.

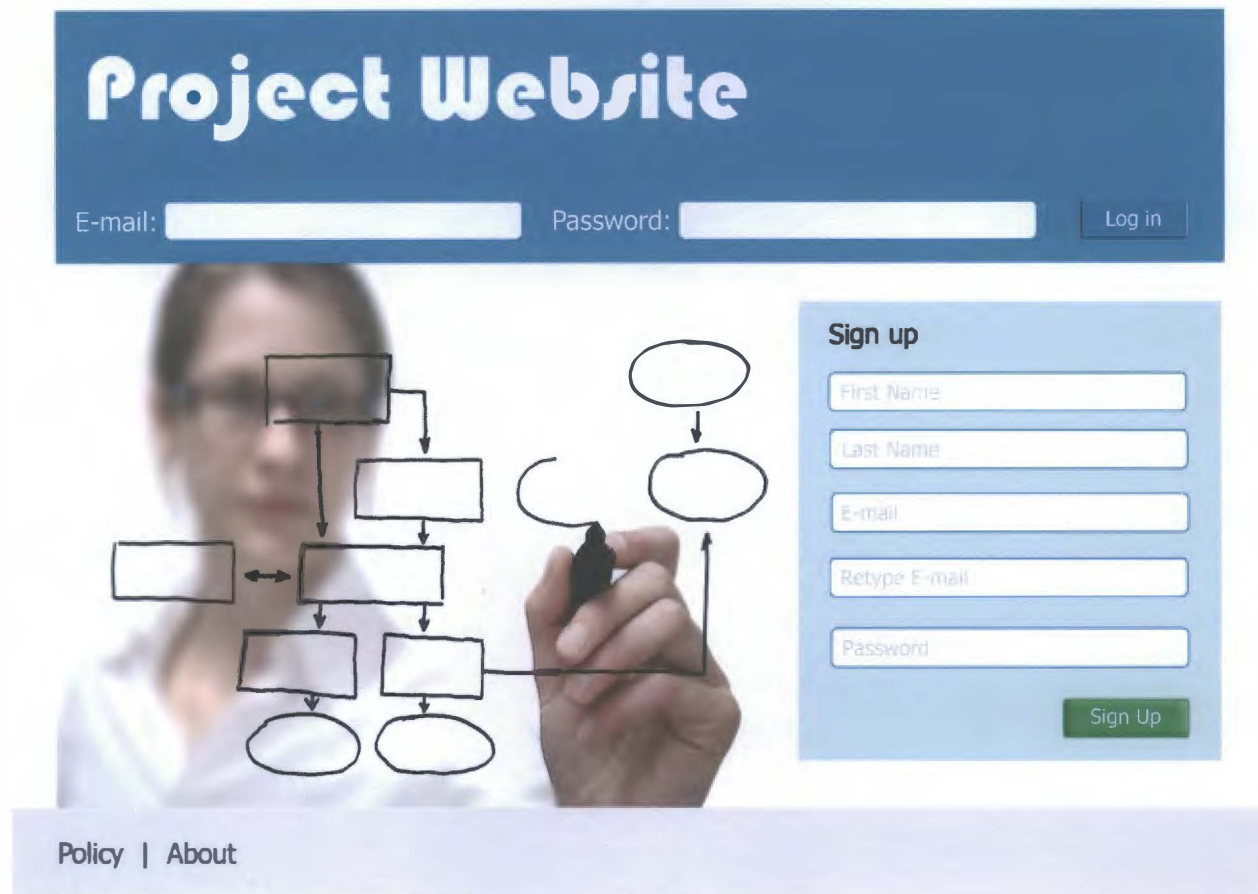


Figure 23 – The initial Photoshop designing for graphic part

This photo is just for the first page, but there are other pages sketched on the paper as they were shown in Figures 17 to 22.

Since the designing phase is a part of this thesis, I am not going to use CMS; instead it is going to be built from the scratch. The easiest way to deal with a bunch of codes in a website is to think about a part of the website that is repeated. Then, create a template and make all pages follow that template. In the following chapter, the implementation of making a PHP template will be explained.

3.3 Backend Storage Design

The database tables are figured out easier with diagrams and flowcharts in Figures 14 to 23. This is important to notice that there two kinds of tables. Resource tables are the

one which are filled out with fix data and by the administrative user. The data saved in these tables are used to fill out other tables' fields. The second group if tables are the ones that are being filled out by registered user in the portal.

3.3.1 Resource Tables

The resource tables should be made first. They do not include any foreign key, but their primary keys will be included in other tables. Tables 1 to 7 show the resource tables.

PRIVACY_TBL would be the first created table. It is a source table that will determine what kind of privacy any of the objects will have.

| | | |
|--------------|---------------|-----------------------|
| PRIVACY_ID | NUMBER | NOT NULL/ PRIMARY KEY |
| PRIVACY_NAME | VARCHAR2(300) | NOT NULL |

Table 1 - Privacy table

INTEREST_AREA_TBL table is another resource table, and it will contain the information about the area of interest of each user.

| | | |
|---------------|---------------|----------------------|
| INTEREST_ID | NUMBER | NOT NULL/PRIMARY KEY |
| INTEREST_NAME | VARCHAR2(25) | NOT NULL |
| EXPLANATION | VARCHAR2(250) | |

Table 2 - Interest area table

MAJOR_TBL contains information about the degree majors. This table will be filled out for users' education data.

| | | |
|----------|--------|----------------------|
| MAJOR_ID | NUMBER | NOT NULL/PRIMARY KEY |
|----------|--------|----------------------|

| | | |
|------------|--------------|----------|
| MAJOR_NAME | VARCHAR2(30) | NOT NULL |
|------------|--------------|----------|

Table 3 - Major table

DEGREE_TBL includes the names of the degrees in general. Because this is a resource table, this will be filled out by fixed data and then will be used by users.

| | | |
|-------------|--------------|----------------------|
| DEGREE_ID | NUMBER | NOT NULL/PRIMARY KEY |
| DEGREE_NAME | VARCHAR2(50) | NOT NULL |

Table 4 - Degree table

SKILL_TBL includes the users' information about their skills to be shown in their profile. The skills' names are fixed though, so no user can add some skills except the ones in the database.

| | | |
|------------|--------------|-----------------------|
| SKILL_ID | NUMBER | NOT NULL/ PRIMARY KEY |
| SKILL_NAME | VARCHAR2(20) | NOT NULL |

Table 5 - Skill table

ROLE_TBL is a table for checking the roles for privacy implementation.

| | | |
|-----------|--------------|-----------------------|
| ROLE_ID | NUMBER | NOT NULL/ PRIMARY KEY |
| ROLE_NAME | VARCHAR2(10) | |

Table 6 - Role table

FEEDBACK_TBL contains the fields of data related to the users' feedbacks.

| | | |
|-------------|---------------|-------------------------|
| FEEDBACK_ID | NUMBER | NOT NULL/PRIMARY KEY |
| STARS_NO | NUMBER (1, 0) | NOT NULL |

Table 7 - Feedback table

3.3.2 Other Tables

Since resource tables need to be made first and might contain some fields of data as the foreign key of others, other tables are different. To prevent SQL PLUS errors in implementation phase, giving priority to the tables that do not have foreign key yet is a significant tip.

USER_TBL is used to record all the basic information of the user. The data types are defined to be clarified for future steps.

| | | |
|-----------|--------------|------------------------|
| USER_ID | NUMBER | NOT NULL / PRIMARY KEY |
| F_NAME | VARCHAR2(30) | NOT NULL |
| L_NAME | VARCHAR2(30) | NOT NULL |
| U_EMAIL | VARCHAR2(50) | NOT NULL |
| U_PASS | VARCHAR2(20) | NOT NULL |
| U_PICTURE | BLOB | |
| active | int | 0 |

Table 8 - User table

DOB_TBL contains information of users' birthdays. The reason that this table is separated from the user table is because of privacy purposes.

| | | |
|------------|--------|------------------------|
| DOB_ID | NUMBER | NOT NULL / PRIMARY KEY |
| DOB | DATE | NOT NULL |
| USER_ID | NUMBER | NOT NULL/ F_KEY |
| PRIVACY_ID | NUMBER | NOT NULL/ F_KEY |

Table 9 - DOB (Date of Birth) table

USER_ROLE_TBL includes all the keys related to the previous tables, like the role table and the user table.

| | | |
|-----------|--------|------------------------|
| U_ROLE_ID | NUMBER | NOT NULL / PRIMARY KEY |
| ROLE_ID | NUMBER | NOT NULL/ F_KEY |
| USER_ID | NUMBER | NOT NULL/ F_KEY |

Table 10 - User-role relation table

USER_LOCATION_TBL saves all the address and location information of the user. Again, the reason for separation of this table from the user table is for privacy. Also, it helps users to save more than one location in their profile.

| | | |
|---------------|--------------|------------------------|
| U_LOCATION_ID | NUMBER | NOT NULL / PRIMARY KEY |
| U_COUNTRY | VARCHAR2(30) | NOT NULL |
| U_STATE | VARCHAR2(20) | NOT NULL |
| U_CITY | VARCHAR2(30) | NOT NULL |
| USER_ID | NUMBER | NOT NULL/ F_KEY |
| PRIVACY_ID | NUMBER | NOT NULL/ F_KEY |

Table 11 - User location table

WORK_TBL contains the information about the job experience of the users. This table is separated from others to reduce data redundancy in situations when a user has more than one work experience, which is in most cases!

| | | |
|--------------|---------------|----------------------|
| WORK_ID | NUMBER | NOT NULL/PRIMARY KEY |
| COMPANY_NAME | VARCHAR2(100) | NOT NULL |
| POSITION | VARCHAR2(50) | NOT NULL |
| WORK_COUNTRY | VARCHAR2(30) | |
| WORK_STATE | VARCHAR2(30) | |
| WORK_CITY | VARCHAR2(50) | |
| S_DATE | DATE | |
| E_DATE | DATE | |
| USER_ID | NUMBER | NOT NULL/ F_KEY |
| PRIVACY_ID | NUMBER | NOT NULL/ F_KEY |

Table 12 - Work table

USER_INTEREST_TBL is another relationship table. It merely contains the foreign keys from previously made tables.

| | | |
|---------------|--------|----------------------|
| U_INTEREST_ID | NUMBER | NOT NULL/PRIMARY KEY |
| USER_ID | NUMBER | NOT NULL/ F_KEY |
| INTEREST_ID | NUMBER | NOT NULL/ F_KEY |

Table 13 - User-interest relationship table

PHONE_TBL contains the users' phone tables. Information is separated in this table for similar reasons as mentioned above.

| | | |
|------------|--------|-------------------------|
| PHONE_ID | NUMBER | NOT NULL/PRIMARY KEY |
| PHONE | NUMBER | NOT NULL |
| USER_ID | NUMBER | NOT NULL/ F_KEY |
| PRIVACY_ID | NUMBER | NOT NULL/ F_KEY |

Table 14 - Phone table

LINK_TBL keeps the social network links or email addresses of the users in case they want to share them.

| | | |
|-----------------|---------------|-------------------------|
| LINK_ID | NUMBER | NOT NULL/PRIMARY KEY |
| LINK | VARCHAR2(100) | NOT NULL |
| USER_LINK_ID | NUMBER | NOT NULL/ F_KEY |
| PRIVACY_LINK_ID | NUMBER | NOT NULL/ F_KEY |

Table 15 - Link table

EDUCATION_TBL saves information about users' education. Most of the fields are coming from the resource tables like the degree and the major tables. In this case, one user can save more than one education history in her/his profile.

| | | |
|--------------|--------|--------------------------|
| EDUCATION_ID | NUMBER | NOT NULL/ PRIMARY KEY |
|--------------|--------|--------------------------|

| | | |
|-------------------|--------------|-----------------|
| S_DATE | DATE | |
| E_DATE | DATE | |
| USER_ID | NUMBER | NOT NULL/ F_KEY |
| DEGREE_ED_ID | NUMBER | NOT NULL/ F_KEY |
| MAJOR_ED_ID | NUMBER | NOT NULL/ F_KEY |
| EDUCATION_COUNTRY | VARCHAR2(30) | |
| EDUCATION_STATE | VARCHAR2(30) | |
| EDUCATION_CITY | VARCHAR2(50) | |
| INSTITUTION_NAME | VARCHAR2(50) | |
| PRIVACY_ID | NUMBER | NOT NULL/ F_KEY |

Table 16 - Education table

USER_SKILL_TBL is a relational table for relating users to their skills.

| | | |
|-------------|--------|-----------------------|
| U_SKILL_ID | NUMBER | NOT NULL/ PRIMARY KEY |
| SKILL_FK_ID | NUMBER | NOT NULL/ F_KEY |
| USER_FK_ID | NUMBER | NOT NULL/ F_KEY |

Table 17 - User-skill relational table

FOLLOWING_TBL contains only foreign keys from the user table that states which users are followers and who they are following.

| | | |
|-------------|--------|-----------------------|
| FOLLOW_ID | NUMBER | NOT NULL/ PRIMARY KEY |
| FOLLOWER_ID | NUMBER | NOT NULL/ F_KEY |
| FOLLOWED_ID | NUMBER | NOT NULL/ F_KEY |

Table 18 - Following table

ENDORSE_TBL keeps the endorsing feedbacks from users who endorse others. This table has two foreign keys from the user table too.

| | | |
|--------------------|-------------|-----------------------|
| ENDORSE_ID | NUMBER | NOT NULL/ PRIMARY KEY |
| ENDORSER_ID | NUMBER | NOT NULL/ F_KEY |
| ENDORSED_ID | NUMBER | NOT NULL/ F_KEY |
| SKILL_ID | NUMBER | NOT NULL/ F_KEY |
| ENDORSE_DATE | DATE | |
| APPROVEMENT_STATUS | VARCHAR2(3) | |

Table 19 - Endorse table

PROJECT_TBL keeps the information about projects.

| | | |
|---------------|---------------|-----------------------|
| PROJECT_ID | NUMBER | NOT NULL/ PRIMARY KEY |
| PROJECT_TITLE | VARCHAR2(50) | NOT NULL |
| EXPLANATION | VARCHAR2(250) | NOT NULL |
| UPDATE_TIME | DATE | |
| PRIVACY_ID | NUMBER | NOT NULL/ F_KEY |
| PROJECT_OWNER | NUMBER | NOT NULL/ F_KEY |

Table 20 - Project table

USER_PROJECT_TBL contains the foreign keys from the user table, the project table, and the role table.

| | | |
|-----------------|--------|----------------------|
| USER_PROJECT_ID | NUMBER | NOT NULL/PRIMARY KEY |
|-----------------|--------|----------------------|

| | | |
|------------|--------|-----------------|
| USER_ID | NUMBER | NOT NULL/ F_KEY |
| PROJECT_ID | NUMBER | NOT NULL/ F_KEY |
| ROLE_ID | NUMBER | NOT NULL/ F_KEY |

Table 21 – User-project relation table

PROJECT_SKILL_TBL is a relational table that keeps foreign keys from the skill table and the project table.

| | | |
|------------------|--------|----------------------|
| PROJECT_SKILL_ID | NUMBER | NOT NULL/PRIMARY KEY |
| SKILL_ID | NUMBER | NOT NULL/ F_KEY |
| PROJECT_ID | NUMBER | NOT NULL/ F_KEY |

Table 22 – Project-skill relation table

PROJECT_INTEREST_TBL is another relational table relating the project table to users' interest areas.

| | | |
|---------------------|--------|----------------------|
| PROJECT_INTEREST_ID | NUMBER | NOT NULL/PRIMARY KEY |
| PROJECT_ID | NUMBER | NOT NULL/ F_KEY |
| INTEREST_ID | NUMBER | NOT NULL/ F_KEY |

Table 23 – Project-interest relation table

STEP_TBL defines steps for every project.

| | | |
|------------|--------------|----------------------|
| STEP_ID | NUMBER | NOT NULL/PRIMARY KEY |
| PROJECT_ID | NUMBER | NOT NULL/ F_KEY |
| STEP_NAME | VARCHAR2(20) | NOT NULL |

| | | |
|--------------------|------|----------|
| SHARE_PERCENT | INT | NOT NULL |
| COMPLETION_PERCENT | INT | NOT NULL |
| UPDATE_DATE | DATE | |
| S_DATE | DATE | NOT NULL |
| E_DATE | DATE | NOT NULL |

Table 24 – Project steps table

STEP_USER_TBL relates the steps to users.

| | | |
|--------------|--------|----------------------|
| STEP_USER_ID | NUMBER | NOT NULL/PRIMARY KEY |
| STEP_ID | NUMBER | NOT NULL/ F_KEY |
| USER_ID | NUMBER | NOT NULL/ F_KEY |

Table 25 – Step-user relation table

FEEDBACK_USER_PROJECT_TBL contains the users’ feedback on each other’s performance in every project.

| | | |
|-------------|--------|----------------------|
| FUP_ID | NUMBER | NOT NULL/PRIMARY KEY |
| FEEDBACK_ID | NUMBER | NOT NULL/ F_KEY |
| GIVER_ID | NUMBER | NOT NULL/ F_KEY |
| RECEIVER_ID | NUMBER | NOT NULL/ F_KEY |
| PROJECT_ID | NUMBER | NOT NULL/ F_KEY |
| PRIVACY_ID | NUMBER | NOT NULL/ F_KEY |

Table 26 – Relational table for users, feedback, and project

IDEA_TBL saves the users' ideas.

| | | |
|----------------|---------------|----------------------|
| IDEA_ID | NUMBER | NOT NULL/PRIMARY KEY |
| IDEA_TITLE | VARCHAR2(20) | NOT NULL |
| IDEA_EXPL | VARCHAR2(300) | NOT NULL |
| IDEA_DATE | DATE | |
| PUBLISH_STATUS | VARCHAR2(3) | NOT NULL |
| INUSE_STATUS | VARCHAR2(3) | NOT NULL |
| USER_ID | NUMBER | NOT NULL/ F_KEY |
| PRIVACY_ID | NUMBER | NOT NULL/ F_KEY |

Table 27 - Idea table

IDEA_PROJECT_TBL is a relational table between ideas and projects.

| | | |
|-----------------|--------|----------------------|
| IDEA_PROJECT_ID | NUMBER | NOT NULL/PRIMARY KEY |
| IDEA_ID | NUMBER | NOT NULL/ F_KEY |
| PROJECT_ID | NUMBER | NOT NULL/ F_KEY |

Table 28 - Idea-project table

IDEA_SKILL_TBL is a relational table for ideas and skills.

| | | |
|---------------|--------|----------------------|
| IDEA_SKILL_ID | NUMBER | NOT NULL/PRIMARY KEY |
| IDEA_ID | NUMBER | NOT NULL/ F_KEY |
| SKILL_ID | NUMBER | NOT NULL/ F_KEY |

Table 29 - Idea-skill relation table

IDEA_INTEREST_AREA is a relation table between the idea table and the interest table.

| | | |
|------------------|--------|----------------------|
| IDEA_INTEREST_ID | NUMBER | NOT NULL/PRIMARY KEY |
| IDEA_ID | NUMBER | NOT NULL/ F_KEY |
| INTEREST_ID | NUMBER | NOT NULL/ F_KEY |

Table 30 – Idea-interest relation table

COMMENT_TBL is a table for saving the comments on projects, ideas, and comments themselves. The foreign keys here can be empty.

| | | |
|-----------------|---------------|----------------------|
| COMMENT_ID | NUMBER | NOT NULL/PRIMARY KEY |
| COMMENT_CONTENT | VARCHAR2(300) | NOT NULL |
| USER_ID | NUMBER | NOT NULL/ F_KEY |
| COMMENT_DATE | DATE | NOT NULL |
| COMMENT_FK | NUMBER | F_KEY |
| IDEA_FK | NUMBER | F_KEY |
| PROJECT_FK | NUMBER | F_KEY |

Table 31 – Comment table

3.3.3 Sample Data

After creating these tables, it is high time to consider and design for the sample data for filling out the tables in the implementation phase (next chapter). Tables 32 to 61 are

showing some planned sample data that will be used during the implementation and testing phase:

| PRIVACY_ID | PRIVACY_NAME |
|------------|--------------|
| 1 | private |
| 2 | public |
| 3 | followers |
| 4 | custom |
| 5 | team |

Table 32 - Privacy sample table

The below table is the interest area table, which is filled out with the information extracted from College Career Life (2008).

| INTEREST_ID | INTEREST_NAME | EXPLANATION |
|-------------|---------------|---|
| 1 | Conventional | Conventional people fit best in organization, structure, and stable communities. They love caring about details in a routine way. |
| 2 | Investigative | Investigative types are interested in solving problems. Science, math, and engineering are the areas that they excel. |
| 3 | Social | Social types care about "helping" through professional relationship like |

| | | |
|---|--------------|--|
| | | nursing, counselling, and teaching. They do enjoy communicating with people. |
| 4 | Artistic | Artistic people highly value creative expression. They enjoy working in an area which does not need rules and regulations. |
| 5 | Realistic | Working outdoor is preferred by realistic. They do not like office tasks and love to work hands-on. |
| 6 | Enterprising | Enterprising people like to carry projects. Taking risks and leadership roles are well-suited for them. |

Table 33 - Interest sample table

The major table is filled out with the majors. There are too many fields that can be filled out, but once they are done, it is worth using. Because of time limitations, it is not filled out completely.

| MAJOR_ID | MAJOR_NAME |
|----------|-------------|
| 1 | History |
| 2 | Linguistics |
| 3 | Arts |

| | |
|----|------------------------------|
| 4 | Philosophy |
| 5 | Religion |
| 6 | Anthropology |
| 7 | Archaeology |
| 8 | Cultural and ethnic studies |
| 9 | Economics |
| 10 | Gender and sexuality studies |
| 11 | Geography |
| 12 | Political science |
| 13 | Psychology |
| 14 | Sociology |
| 15 | Biology |
| 16 | Chemistry |
| 17 | Physics |
| 18 | Space sciences |
| 19 | Mathematics |
| 20 | Computer sciences |
| 21 | Statistics |
| 22 | Business |
| 23 | Education |
| 24 | Agriculture |
| 25 | Architecture and design |

| | |
|----|------------------------|
| 26 | Engineering |
| 27 | Law |
| 28 | Medicine |
| 29 | Information Technology |

Table 34 - Major sample table

According to Study (n.d.), the list of main degrees that are used globally are as

follows:

| DEGREE_ID | DEGREE_NAME |
|-----------|---------------|
| 1 | Undergraduate |
| 2 | Associate's |
| 3 | Bachelor's |
| 4 | Master's |
| 5 | Doctoral |

Table 35 - Degree sample table

The numbers of skills that a person can have varies, but in Example Your Dictionary, it is cited as fixed data. The skill table below contains just a very small number of them.

| SKILL_ID | SKILL_NAME |
|----------|--------------------------------|
| 1 | Ability to work under pressure |
| 2 | Accuracy |
| 3 | Adaptability |
| 4 | Administering medication |

| | |
|---|--------------------|
| 5 | Advising people |
| 6 | Analyzing data |
| 7 | Analyzing problems |

Table 36 - Skill sample table

The role table has the resource data like this:

| ROLE_ID | ROLE_NAME |
|---------|-----------|
| 1 | member |
| 2 | leader |
| 3 | owner |
| 4 | admin |

Table 37 - Role sample table

The example data for the feedback table as one of the resource tables look like this:

| FEEDBACK_ID | STARS_NO |
|-------------|----------|
| 1 | 1 |
| 2 | 2 |
| 3 | 3 |
| 4 | 4 |
| 5 | 5 |

Table 38 - Feedback sample table

The user table is the first table that is not a resource table. The examples of users' information are saved in this table.

| USER_ID | F_NAME | L_NAME | U_EMAIL | U_PASSWORD | U_PICTURE | active |
|---------|-------------|----------------|----------------------------|------------|-----------|--------|
| 1 | Fateme h | Khastkhod a | fkhastkhodaard@eiu.ed u | fA1# | | 1 |
| 2 | Mark | Zuckerber g | mzuckerberg@fb.com | fAce@ 2 | | 1 |
| 3 | Larry | Page | lpage@gmail.com | LaRp# | | 1 |

Table 39 - User table samples

The information entered here will be implemented in the next chapter.

| U_ROLE_ID | ROLE_ID | USER_ID |
|-----------|---------|---------|
| 1 | 1 | 1 |
| 2 | 2 | 1 |
| 3 | 1 | 2 |
| 4 | 1 | 3 |
| 5 | 3 | 2 |
| 6 | 1 | 5 |

Table 40 - User/role relationship table

A sample of users' birthdays are as follows:

| DOB_ID | DOB | USER_ID | PRIVACY_ID |
|--------|------------|---------|------------|
| 1 | 04/03/1988 | 1 | 3 |
| 2 | 05/06/1981 | 2 | 1 |

| | | | |
|---|------------|---|---|
| 3 | 10/11/1986 | 3 | 2 |
|---|------------|---|---|

Table 41 - Users' date of birth

Users' locations are considered to be as follows:

| U_COUNTRY_ID | U_COUNTRY | U_STATE | U_CITY | U_ID | PRIVACY_ID |
|--------------|---------------|---------|---------|------|------------|
| 1 | United States | WA | Seattle | 2 | 2 |
| 2 | Iran | Fars | Shiraz | 1 | 3 |
| 3 | United States | IL | Chicago | 3 | 4 |

Table 42 - Users' locations

Users' work experiences are designed to have this information:

| WORK_ID | COMPANY_NAME | POSITION_NAME | WORK_COUNTRY | WORK_STATE | WORK_CITY | START_DATE | END_DATE | USER_ID | PRIVACY_ID |
|---------|--------------|--------------------|--------------|------------|-----------|------------|------------|---------|------------|
| 1 | Deka Steel | Project controller | Iran | Fars | Shiraz | 7/15/2012 | 10/15/2012 | 1 | 2 |
| 4 | facebook | manager | us | CA | M | 1/1/2000 | now | 2 | 5 |
| 5 | facebook | marketing | us | CA | M | 1/4/2008 | now | 2 | 5 |

Table 43 - Users' work experience

Users' interest areas contain these examples to be tested in the next chapters.

| | | |
|------------------|---------|-------------|
| USER_INTEREST_ID | USER_ID | INTEREST_ID |
|------------------|---------|-------------|

| | | |
|---|---|---|
| 1 | 1 | 3 |
| 2 | 1 | 4 |

Table 44 – User and interest areas’ relationship table

The phone table is considered to contain these phone numbers in them.

| PHONE_ID | PH_NUM | USER_ID | PRIVACY_ID |
|----------|---------------|---------|------------|
| 1 | 0012172541879 | 1 | 1 |
| 2 | 0018002541879 | 2 | 2 |
| 3 | 0019012541879 | 3 | 2 |

Table 45 – Phone table examples

The link table saves some external users’ links in them.

| LINK_ID | LINK | USER_ID | PRIVACY_ID |
|---------|---|---------|------------|
| 1 | http://iransetarehgasht.com/ | 1 | 2 |
| 2 | www.google.com | 2 | 2 |
| 3 | www.facebook.com | 3 | 2 |

Table 46 – Link table samples

Users’ educations are saved as follows:

| EDUCATION_ID | S_DATE | E_DATE | USER_ID | DEGREE_ID | MAJOR_ID | DEGREE_COUNTRY | DEGREE_STATE | DEGREE_CITY | DEGREE_INSTITUTION | PRIVACY_ID |
|--------------|--------|--------|---------|-----------|----------|----------------|--------------|-------------|--------------------|------------|
|--------------|--------|--------|---------|-----------|----------|----------------|--------------|-------------|--------------------|------------|

| | | | | | | | | | | |
|---|---------------|----------------|---|---|----|------|------|------------|--|---|
| 1 | 6/1/ 2006 | 11/1/ 2011 | 1 | 3 | 29 | Iran | Fars | Shiraz | Fars University Science & Research Branch | 2 |
| 2 | 8/25 /2014 | 12/19 /2015 | 1 | 4 | 29 | us | IL | Charleston | EIU | 2 |

Table 47 - Users' sample of educations

The relational table between the user and skills tables have these examples for testing:

| USER_SKILL_ID | SKILL_ID | USER_ID |
|---------------|----------|---------|
| 1 | 1 | 1 |
| 2 | 2 | 1 |
| 3 | 3 | 1 |

Table 48 - User/skills relationship examples

The following table is designed to be filled out like this:

| FOLLOW_ID | FOLLOWER_USER_ID | FOLLOWED_USER_ID |
|-----------|------------------|------------------|
| 1 | 1 | 2 |
| 2 | 1 | 3 |

| | | |
|---|---|---|
| 3 | 2 | 1 |
| 4 | 3 | 2 |

Table 49 - Following table with sample data

Based on previous tables, the endorse table is designed to save this data.

| ENDORSE_ID | ENDORISING_USER | ENDORSED_USER | SKILL_ID | DATE | APPROVEMENT_STATUS |
|------------|-----------------|---------------|----------|-----------|--------------------|
| 1 | 1 | 2 | 1 | 1/5/2015 | YES |
| 2 | 2 | 3 | 2 | 7/5/2015 | NO |
| 3 | 2 | 1 | 3 | 11/5/2015 | YES |
| 4 | 3 | 2 | 3 | 1/7/2015 | YES |

Table 50 - Endorsement table with sample data

The project table contains some sample data for the implementation phase.

| PROJECT_ID | PROJECT_TITLE | EXPLANATION | UPDATE_TIME |
|------------|------------------------|--|-------------|
| 1 | Project Website | The project is about a website.... | 7/1/2015 |
| 2 | Emoji Effects on Users | This research is about the effects of emoji on users | 10/8/15 |

| | | | | | | | |
|---|---|----------------|----|----|---------|----------|----------|
| 1 | 1 | Researching | 10 | 50 | 7/1/15 | 6/1/15 | 12/1/15 |
| 2 | 1 | Designing | 30 | 70 | 10/8/15 | 9/1/15 | 10/15/15 |
| 3 | 1 | Implementation | 30 | 0 | 10/8/15 | 10/15/15 | 10/30/15 |
| 4 | 1 | Testing | 20 | 0 | 10/8/15 | 10/15/15 | 10/30/15 |
| 5 | 1 | maintenance | 10 | 20 | 10/8/15 | 11/1/15 | 1/1/16 |

Table 54 - The projects' steps samples

The following table contains the samples for the relationship table between the user and project steps tables.

| STEP_USER_ID | STEP_ID | USER_ID |
|--------------|---------|---------|
| 1 | 1 | 1 |
| 2 | 2 | 1 |

Table 55 - The relationship data sample for users and steps

The FEEDBACK_USER_PROJECT_TBL contains five foreign keys, and the following sample data comes from previously made tables.

| FUP_ID | FEEDBACK_ID | USER_GIVING_ID | USER_RECEIVING_ID | PROJECT_ID | PRIVACY_ID |
|--------|-------------|----------------|-------------------|------------|------------|
| D | D | D | D | D | D |

| | | | | | |
|---|---|---|---|---|---|
| 1 | 4 | 2 | 1 | 2 | 2 |
| 2 | 5 | 1 | 2 | 2 | 2 |

Table 56 - The relationship sample data for feedbacking users in projects

The IDEA_TBL is supposed to have the following data samples for the implementation phase if needed.

| IDEA_ID | IDEA_TITL | IDEA_EXPL | IDEA_DATE | PUBLISH_STATU | INUSE_STATU | USER_ID | PRIVACY_ID |
|---------|-----------|------------------|-----------|---------------|-------------|---------|------------|
| 1 | Project | The | 1/1/201 | yes | yes | 1 | 4 |
| | Website | bjsjnsjnscn c | 5 | | | | |

Table 57 - Idea table sample

The following sample data is saved in IDEA_PROJECT_TBL as follows.

| IDEA_PROJECT_ID | IDEA_ID | PROJECT_ID |
|-----------------|---------|------------|
| 1 | 1 | 1 |

Table 58 - Idea/project sample data

The IDEA_SKILL_TBL is designed to have the following data, as an example.

| IDEA_SKILL_ID | IDEA_ID | SKILL_ID |
|---------------|---------|----------|
| 1 | 1 | 1 |

Table 59 - Idea/skill relationship sample

The IDEA_INTEREST_AREA has the following data as an example.

| IDEA_INTEREST_ID | IDEA_ID | INTEREST_ID |
|------------------|---------|-------------|
| 1 | 1 | 1 |

Table 60 - Idea/interest example

The comment table contains the following example.

| COMMENT_ID | COMMENT | USER_ID | DATE | PARENT |
|------------|----------|---------|---------|----------------|
| 1 | Good job | 2 | 10/8/15 | PROJECT_TBL(1) |

Table 61 - The comment table example

Now is the time to start the implementation phase for the database. In the next chapter, programming and codes will be discussed.

Chapter 4: Implementation Phase

4.1 Installing Oracle Database and XAMPP Server

For this project, the best set of software that I found reasonable to install was **Oracle Enterprise 11g Release 2 with XAMPP server**. I chose Enterprise Version because it has special features for security, and XAMPP Server because I could connect it to Oracle. Both of them are free, and there is no need to pay for the software.

Oracle Enterprise is bulky software; once installed on Operating System (OS), it is hard to completely be removed. For installing Oracle and not damaging the OS, the best approach is to install a virtual machine. There are different versions of VMware that one can use.

The question is what is VMware? It is a virtual machine to simulate an OS in a limited space, so that one can test the software without damaging the system. It is recommended to all people who want to install new and challenging software. Among all the versions, there are two that got my attention. The Workstation version is not free, but it is a very good option for those who want to install different kinds of OS and try to make a team or network. The Player version is free, and it is a good option for students and people who do not need to connect different OS together. If one has decided to install this, she/he has to consider its different versions for Mac or Windows.

Depending on the OS, one can find the installation file on the Oracle website. After downloading the Oracle files (they are two files usually), installation will begin. A good video tutorial that has explained all about it will be very beneficial.

The next step would be installing XAMPP Server, which is pretty easy, by downloading it from Apache Friends (2015).

4.2 Connecting XAMPP Server and Oracle Database

After installing the above software, it is the time to connect XAMPP Server and Oracle. For connecting these two, there are some points that should be noticed. First of all, we need to check to see if everything is working or not:

1. Make sure that the Oracle is working. For doing that we should login to the SQL PLUS by entering the username and password for the “system” user. The password is something that you have entered during the Oracle installation. If connected we will be sure that we can move to the next step.
2. Make sure that the Oracle Listener Service is started by going to Control Panel, Services, and find something like “OracleOraDb11gTNSListener.” If the status was “started” then it is okay, otherwise it should be started manually, but it is usually started by default.
3. In the SQL PLUS, type “tnsping SIDname.” In my case, I typed: “tnsping orcl”

This should give the service name and all the information needed for the next steps.

Write down the service name; it will be needed later. Then, the following changes need to be made in XAMPP files.

1. Download Oracle Instant Client. Its version depends on the Oracle’s version. In my case, I have downloaded Oracle Instant Client for oracle11g-release2.
2. Extract Oracle Instant Client to C:\instantclient_11_2 folder.
3. Open system settings at Start Menu > My Computer (right click on the icon) > select Properties > Advanced Tabs > Environment Variables.
4. Add to ‘System Variables’ by clicking ‘New’ button and insert the following values:
 - a. Variable Name: PATH

b. Variable Value: c:\instantclient_11_2

If there is a need to add two paths, one can put (;) symbol at the end of the first path and without any space put the second path right after.

5. Click OK to accept new modification.
6. Open C:\xampp\php\php.ini by using Notepad.
7. Find string "oci8" and remove the (;) symbol at the beginning of the string line.
8. Save and Exit.
9. Restart the laptop/PC to accept the new modification on XAMPP.

For the next steps, one should make a page to start PHP programming. For making the first page, one can follow the below steps:

1. Go to the XAMPP file, which should be in C:\xampp by default. Double click on "htdocs."
2. Right click > New > Text Document
3. Copy the code lines from Figure 24.

```
<!DOCTYPE html>

<html>

<body>

<?php

    //testing to see if the connection to the Oracle works or not!

    $conn = oci_connect("username","password","localhost/service-name");
```

```
        If (!$conn)
            echo 'Failed to connect to Oracle';
        else
            echo "Successfully connected with Oracle DB :-)";
?>

</body>
</html>
```

Figure 24 – General code for PHP/Oracle connection

Username and password would be those of a user in Oracle. The default username that all use is “system,” but it is better to avoid that because of security. The better option would be to activate one of the users in the Oracle and try to connect to that. In my case, I tried to connect to “PROJECT” with the “project” password. Based on Figure 24, the localhost can stay like that, but the service name varies (which is explained above).

4. Go to File > Save As > name the file something like “connection.php.”
5. Once saved, open a browser and enter this URL: <http://localhost/connection.php>
6. If you see a message stating “Successfully connected with Oracle DB :-) ,” then you are good to go.

Figure 25 shows the real connection code in this project.

```
<?php
$c=oci_connect("project", "project", "localhost/orcl.16.173.130")
If (!$c){
    echo "Oracle Connect Error " ;
} else {
echo "Successfully connected with Oracle DB :-)";
}
?>
```

Figure 25 – Specific coding for PHP/Oracle connection

4.3 Creating PHP Template

As mentioned in chapter three, when there is a repetitive pattern, coders are better to design a template and make all the pages follow that template instead of making them one by one. Therefore, I made my template before going to the database side.

For this purpose, I used CSS, JavaScript, and mainly PHP to make the template work. The index page did not need PHP for creating its template because it was only one page, and there was no need for making similar pages. However, the CSS codes for the index page and the profile pages were similar, which is completely dependent on designer's taste. Figures 26 and 27 show the index and profile pages as outside and inside of the website designing, respectively.

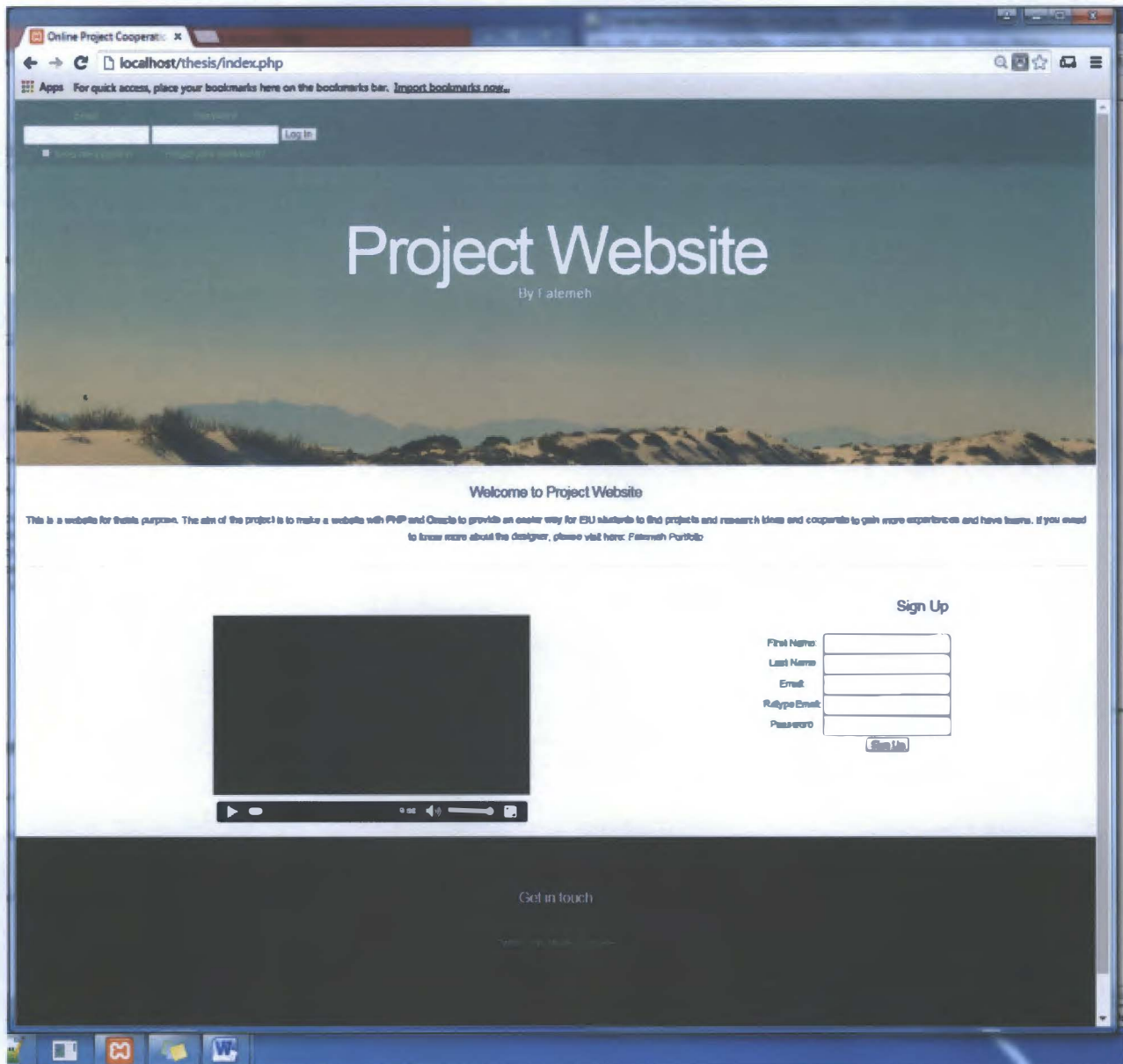


Figure 26 – The index page of the website

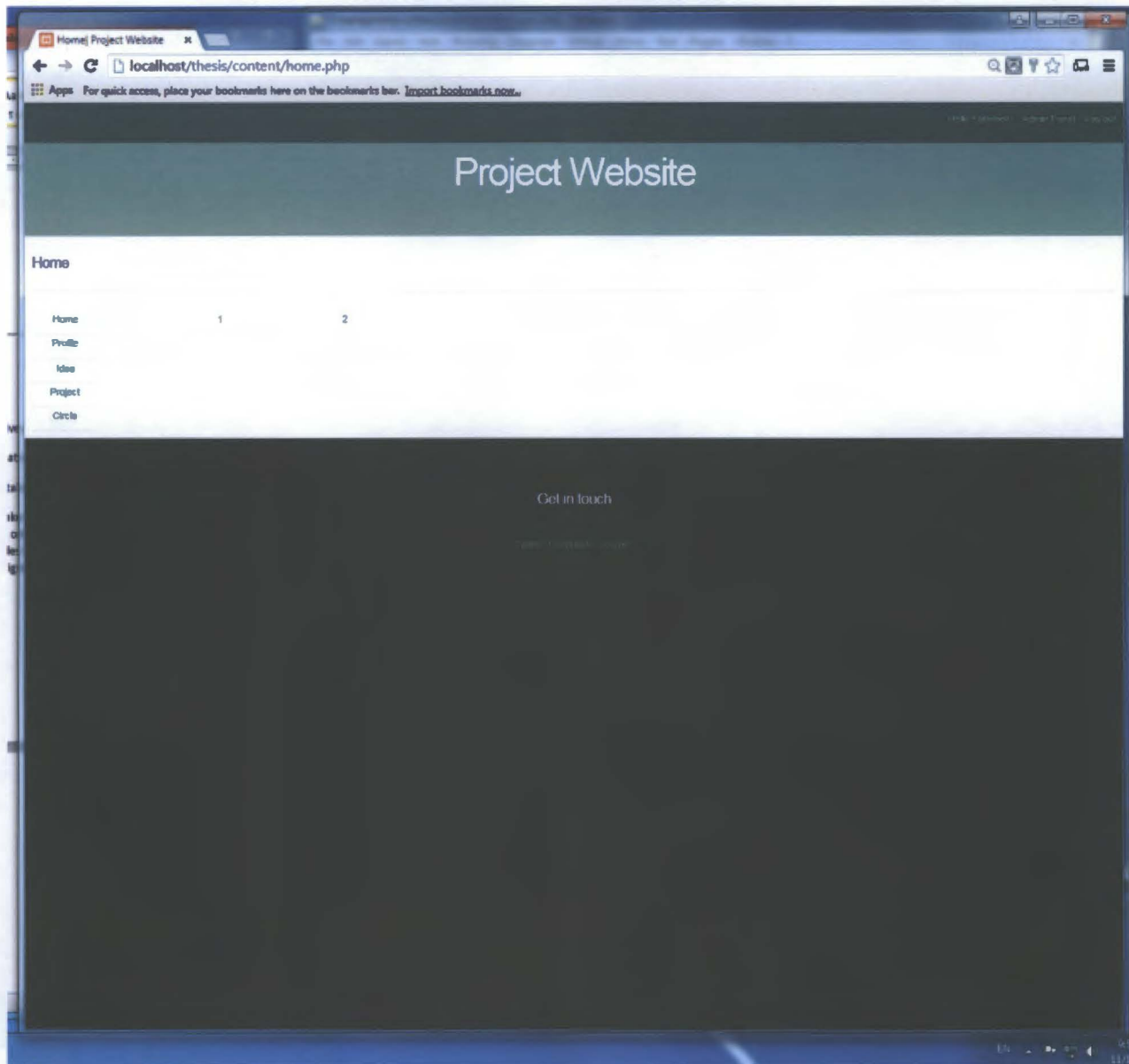


Figure 27 – Home page after login

The CSS and JS files and codes are shown in Figures 28, 29, and 30. The JS file is just used in the index page and for the sign up form to check validation of the fields.

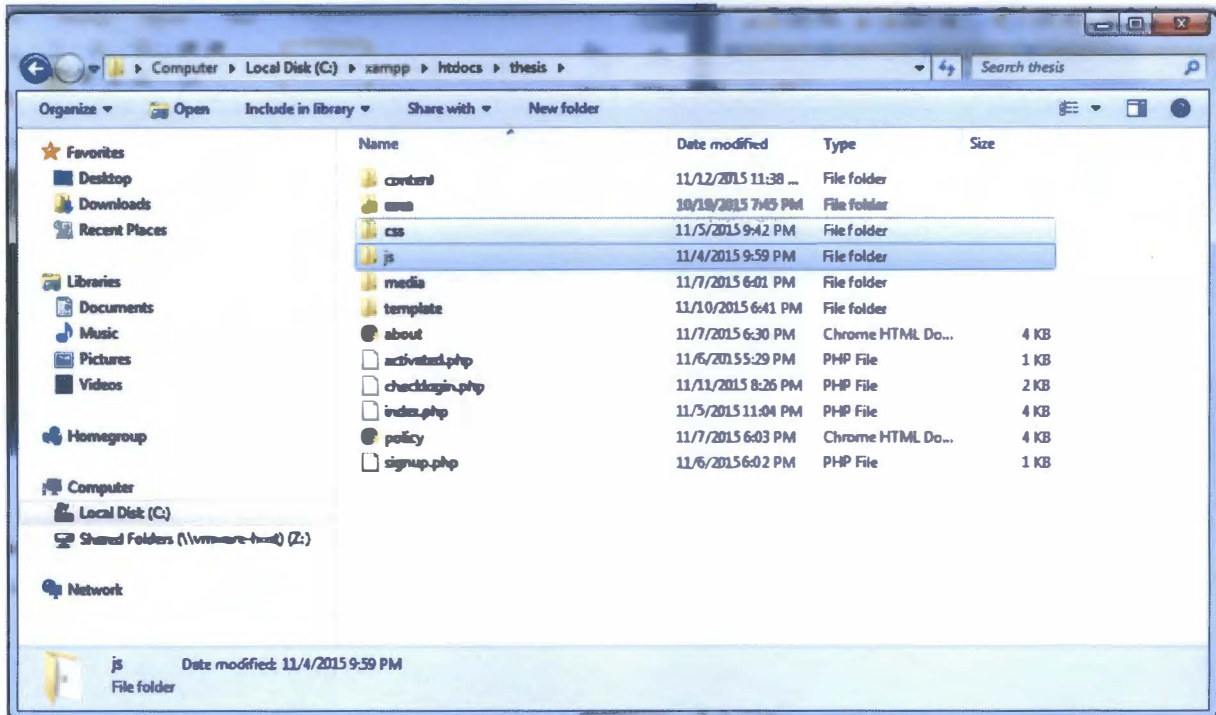
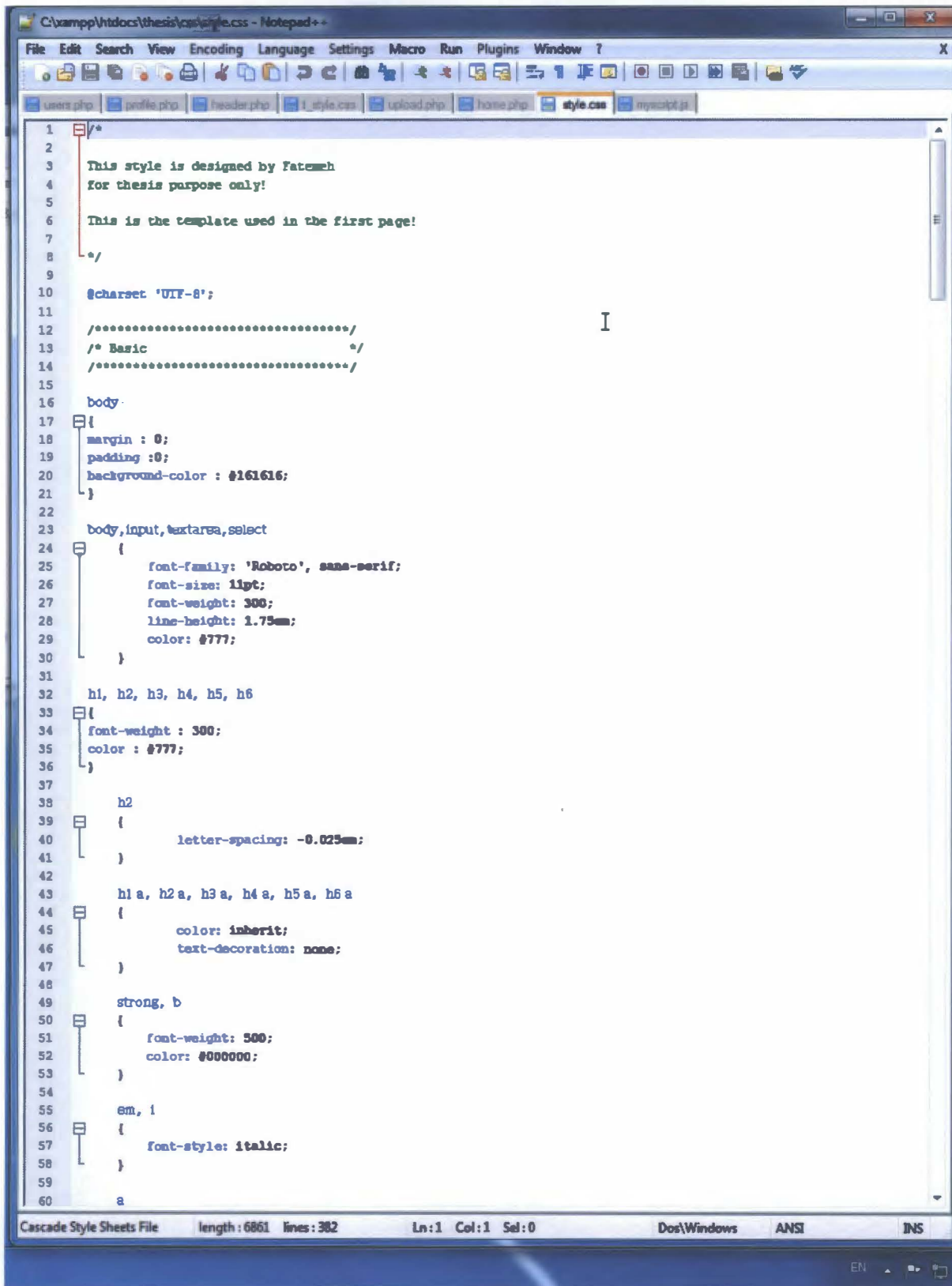


Figure 28 – CSS and JS files in the website root



```
1 /*
2
3 This style is designed by Fatemeh
4 for thesis purpose only!
5
6 This is the template used in the first page!
7
8 */
9
10 @charset 'UTF-8';
11
12 /*****
13  /* Basic
14  *****/
15
16 body
17 {
18     margin : 0;
19     padding :0;
20     background-color : #161616;
21 }
22
23 body, input, textarea, select
24 {
25     font-family: 'Roboto', sans-serif;
26     font-size: 11pt;
27     font-weight: 300;
28     line-height: 1.75em;
29     color: #777;
30 }
31
32 h1, h2, h3, h4, h5, h6
33 {
34     font-weight : 300;
35     color : #777;
36 }
37
38 h2
39 {
40     letter-spacing: -0.025em;
41 }
42
43 h1 a, h2 a, h3 a, h4 a, h5 a, h6 a
44 {
45     color: inherit;
46     text-decoration: none;
47 }
48
49 strong, b
50 {
51     font-weight: 500;
52     color: #000000;
53 }
54
55 em, i
56 {
57     font-style: italic;
58 }
59
60 a
```

Figure 29 – A part of CSS code

```

C:\xampp\htdocs\thesis\myproject.js - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
users.php profile.php header.php l_style.css upload.php home.php style.css myscript.js
1 function confirmEmail() {
2     var email = document.getElementById("email").value;
3     var confemail = document.getElementById("confemail").value;
4     if(email != confemail) {
5         alert('Email Not Matching!');
6         document.getElementById("confemail").value="";
7     }
8 }
9
10 function checkForm(form)
11 {
12     re = /^[a-zA-Z]+$/;
13     if(!re.test(form.fname.value)) {
14         alert("Error: First name must contain only letters!");
15         form.fname.focus();
16         return false;
17     }
18     re = /^[a-zA-Z]+$/;
19     if(!re.test(form.lname.value)) {
20         alert("Error: Last name must contain only letters!");
21         form.lname.focus();
22         return false;
23     }
24     if(form.pass.value == form.email.value) {
25         alert("Error: Password must be different from Username!");
26         form.pass.focus();
27         return false;
28     }
29     re = /[0-9]/;
30     if(!re.test(form.pass.value)) {
31         alert("Error: password must contain at least one number (0-9)!");
32         form.pass.focus();
33         return false;
34     }
35     re = /[a-z]/;
36     if(!re.test(form.pass.value)) {
37         alert("Error: password must contain at least one lowercase letter (a-z)!");
38         form.pass.focus();
39         return false;
40     }
41     re = /[A-Z]/;
42     if(!re.test(form.pass.value)) {
43         alert("Error: password must contain at least one uppercase letter (A-Z)!");
44         form.pass.focus();
45         return false;
46     }
47
48     alert("You entered a valid password: " + form.pwd1.value);
49     return true;
50 }
51
52
JavaScript file      length:1532 lines:52      Ln:1 Col:1 Sel:0      Dos\Windows  ANSI  INS

```

Figure 30 – The whole used JavaScript code

One of the great new features of HTML that has been added to HTML5 is the power of validating email without using JavaScript. This technology is used in my login form and works properly. Figure 31 shows the error when email validation is used merely by HTML coding. In this example, the validation check is not only limited to check for the "@" sign, but also it is fairly accurate. All the CSS and JS codes are in Appendix A.

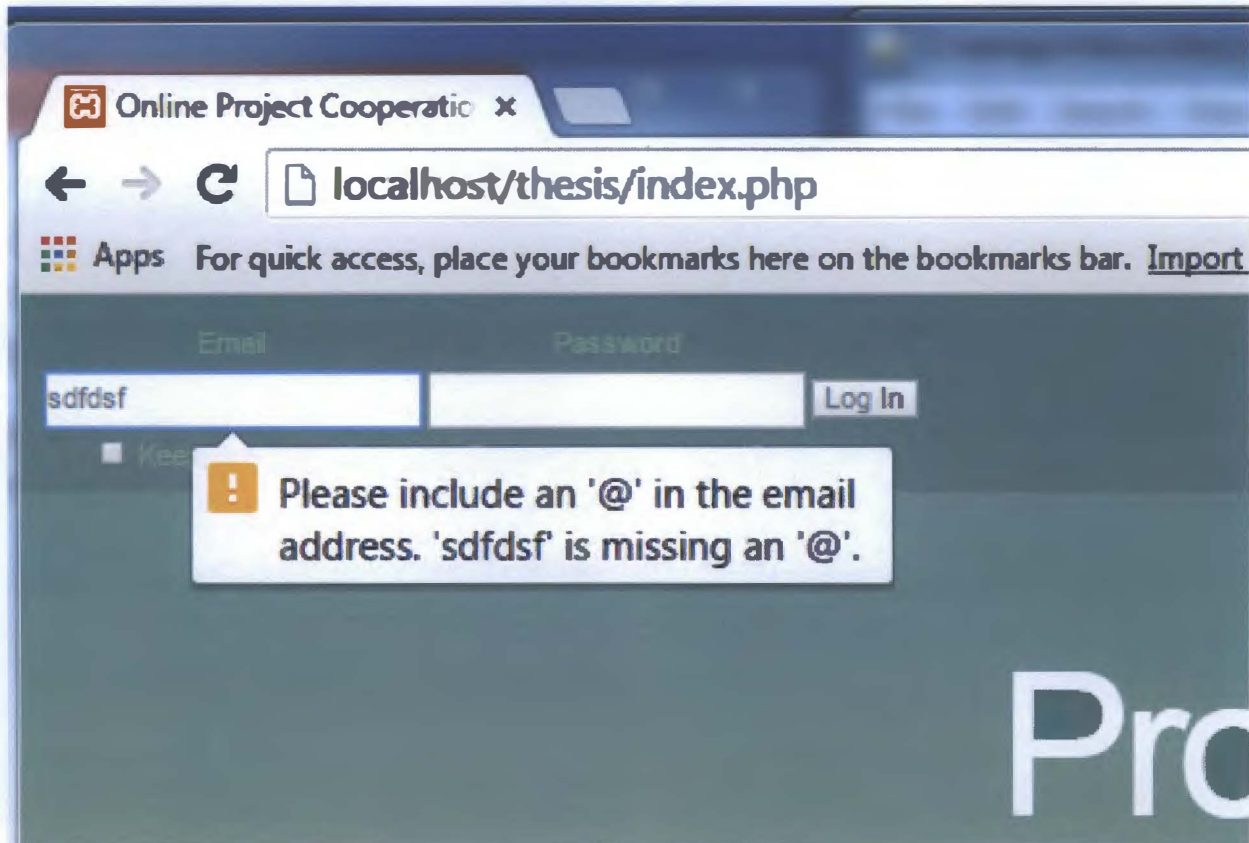


Figure 31 – use of HTML5 email validation

Inside the profile and when the user is logged in, the real PHP template is working. Creating a template is not a hard process as long as one can understand the PHP “required” and “including” commands. Figure 32 and 33 shows the real files which are playing the main role in the PHP templates. All the PHP codes related to the template are in Appendix B.

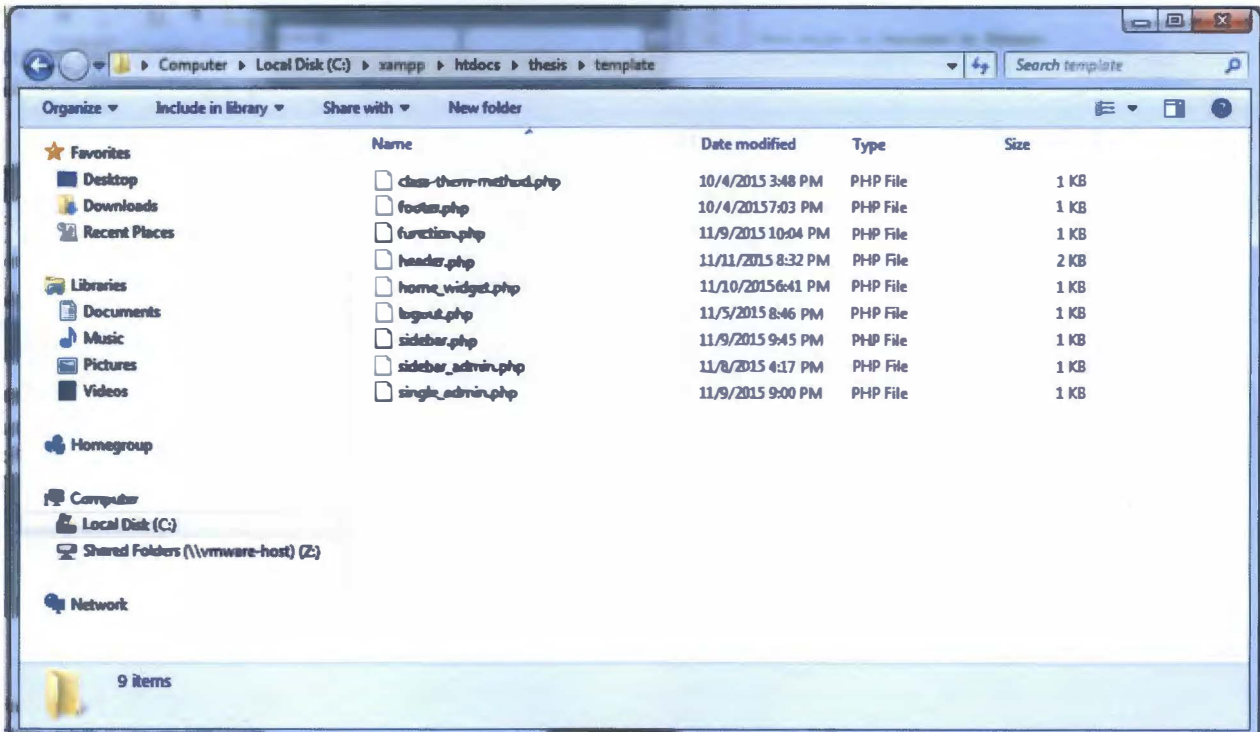


Figure 32 - The template files which are read by pages

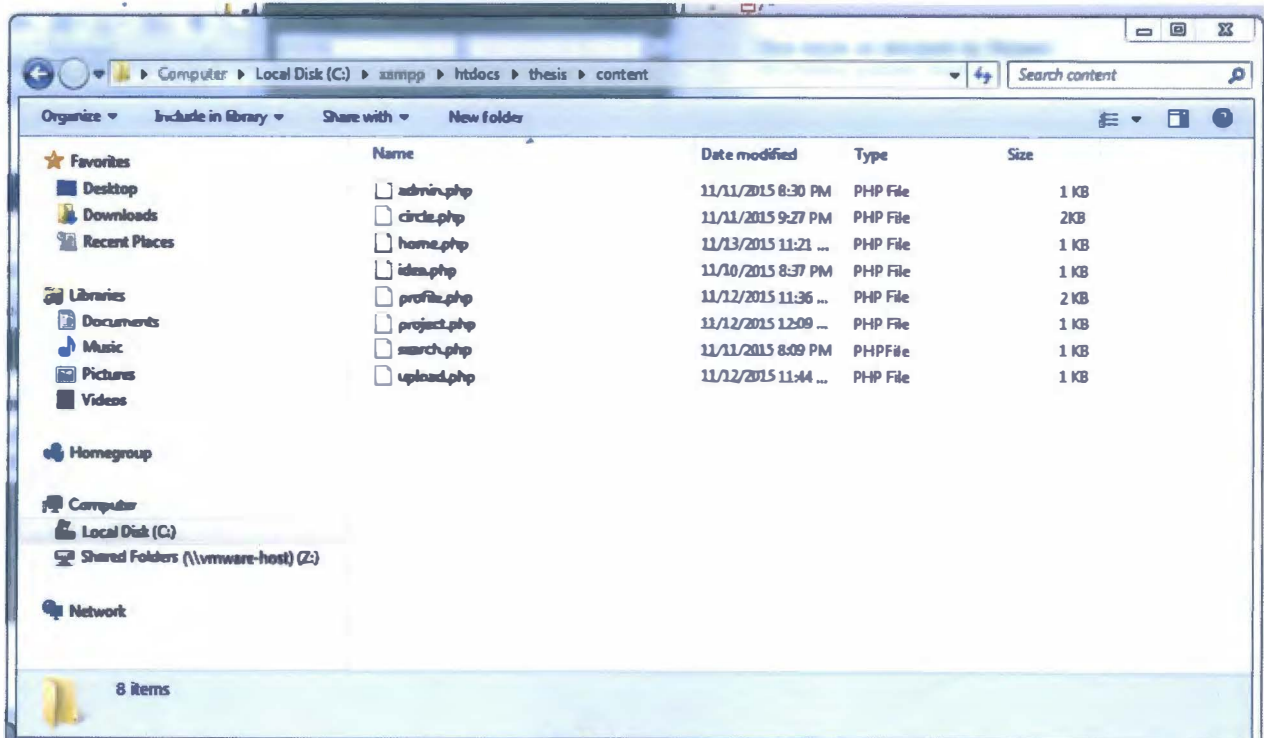


Figure 33 - The pages which are reading the template files

For instance, the header.php is shown in Figure 34.

```

1  <?php
2  include('../template/function.php');
3  ?>
4  <DOCTYPE html>
5  <html>
6  <head>
7  <title><?php echo do_html_title($the_title);?></title>
8  <link rel="stylesheet" type="text/css" href="css/t_style.css" />
9  </head>
10 <body>
11 <div id="header">
12 <div id="nav-wrapper">
13 <!-- Nav -->
14 <nav id="nav">
15
16 <table>
17 <tr>
18 <td>Hello <?php
19 <td>
20     if(logged_in()===true){
21         $r= get_data($session_id);
22         print_r($r['F_NAME'][0]);
23     } else{
24         header("location: ../index.php");
25     }
26 </td>
27 </tr>
28 <tr><?php if(is_admin($session_id) === true){
29 echo '<a href="content/admin.php">Admin Panel</a>';
30 } ?> <td> </tr>
31 </table>
32 <tr><td><a href="template/layout.php">Log out</a>
33 </td>
34 </tr>
35 </table>
36
37 </nav>
38 </div>
39 <div>
40
41 <!-- Logo -->
42 <div id="logo">
43 <br>
44 <br>
45 <br>
46
47
48 <a href="#">Project Website</a></div>
49
50 </div>
51 </div>
52 </div>
53

```

I

PHP Hypertext Preprocessor file length: 1076 lines: 53 Ln: 53 Col: 5 Sel: 0 Dos\Windows ANSI INS

Figure 34 – The header.php codes

As it is obvious, the written codes in this PHP file are not the only codes which are working. The command “include” at the very first line of PHP codes reads another PHP file that without it the header.php cannot work properly. Besides that, there are three important PHP functions that are called to print the title of that page, get the session id, and check if the user is admin or not. All these functions are placed in other files named function.php and users.php. Creating other template pages would need at least one of these functions.

Another example for one of the template pages after log in is shown in Figure 35.

```

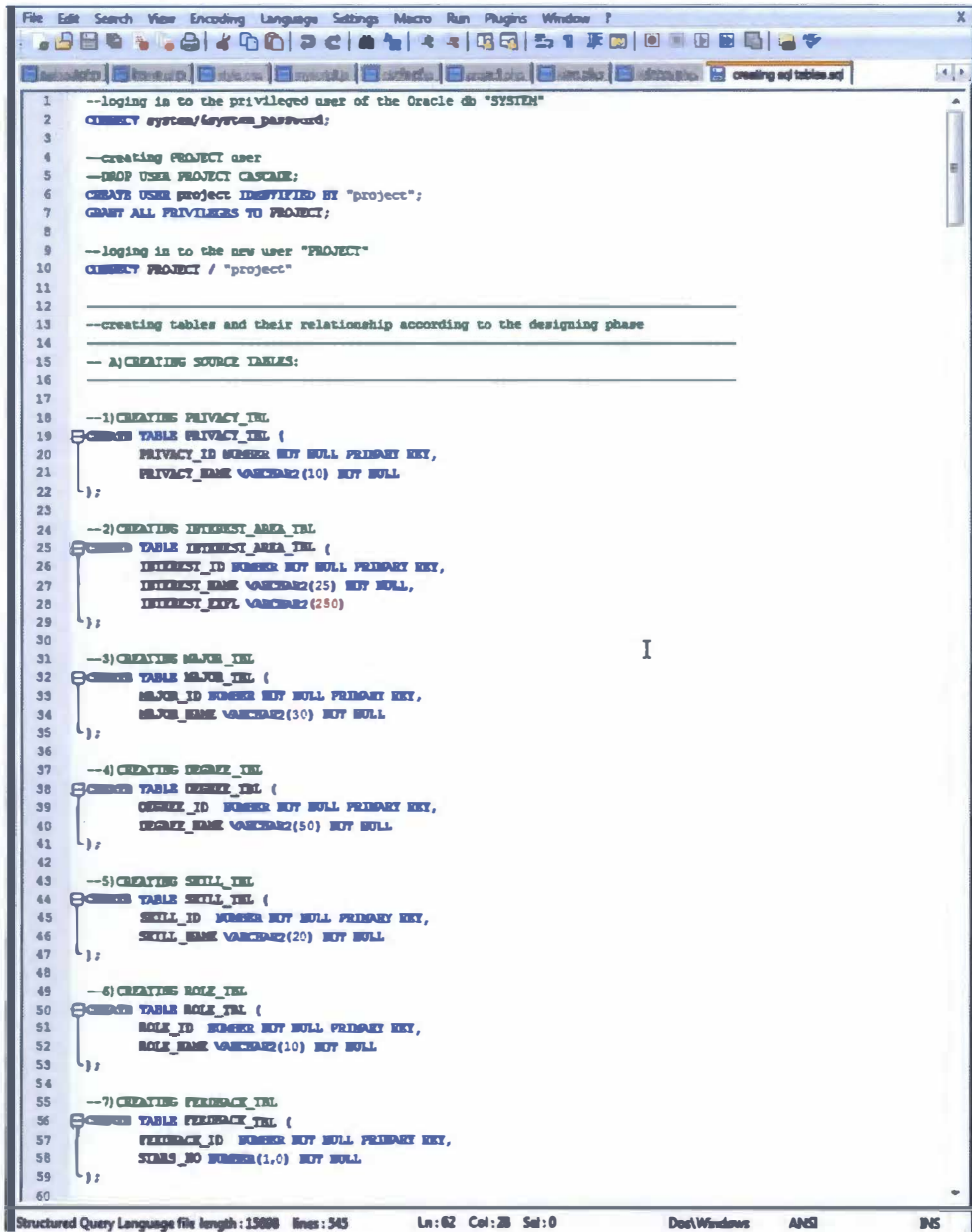
C:\xampp\htdocs\thesis\content\circle.php - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
class them-method.php admin.php circle.php login.php
1 <?php
2 $the_title = 'Circle';
3 include('../core/init.php');
4 include('../template/header.php');
5
6 ?>
7 <div id="featured">
8
9 <header><h2>Circle</h2></header>
10 <hr />
11 <div class="row">
12 <section class="2u" id="sidebar">
13 <?php include('../template/sidebar.php');
14 ?>
15 </section>
16 <section class="8u">
17 <table> <tr>
18 <td><p> You have <?php echo count_circle($session_id); ?> people in your circle !</p></td>
19 <td> <?php echo count_circle($session_id); ?> people in your circle !</td>
20 <td>Find more people: </td>
21 <td> <form method="get" action="search.php" id="searchform">
22 <input type="email" name="email" placeholder="Enter your friend's e-mail">
23 <input type="submit" name="submit" value="Search">
24 </form> </td>
25 </table>
26 <hr />
27 <h4> Followers:</h4>
28 <div class="row">
29 <?php
30 $r=follower($session_id);
31 $n= count($r);
32
33 foreach ($r as $row){
34 echo '<section class="2u">';
35 echo '<a href="#">' . $row . '</a>';
36 echo '</section>';}
37
38 ?>
39 </div>
40
41
42 <hr />
43 <h4> Following:</h4>
44 <div class="row">
45 <?php
46
47
48 $r=followed($session_id);
49 $n= count($r);
50
51
52 foreach ($r as $row){
53 echo '<section class="2u">';
54 echo '<a href="#">' . $row . '</a>';
55 echo '</section>';
56 }
57
58
59 ?>
60 </div>
PHP Hypertext Preprocessor file length: 1416 lines: 67 Ln:11 Col:27 Sel:0 Dot\Windows ANSI INS
8:50 AM
11/13

```

Figure 35 - Circle.php codes

4.4 Creating and Populating Tables

Creating and populating Oracle tables can be done along with PHP designing. There is no priority between any of these phases. A good designing is really beneficial in this phase though. Thus, based on the designing phase, we can create the tables and populate them. Figure 36 shows the resource table creation codes.



```
1 --logging in to the privileged user of the Oracle db "SYSTEM"
2 CONNECT system/system_password;
3
4 --creating PROJECT user
5 --DROP USER PROJECT CASCADE;
6 CREATE USER project IDENTIFIED BY "project";
7 GRANT ALL PRIVILEGES TO PROJECT;
8
9 --logging in to the new user "PROJECT"
10 CONNECT PROJECT / "project"
11
12
13 -----
14 -----
15 -- A) CREATING SOURCE TABLES:
16 -----
17 -----
18 --1) CREATING PRIVACY_TBL
19 CREATE TABLE PRIVACY_TBL (
20     PRIVACY_ID NUMBER NOT NULL PRIMARY KEY,
21     PRIVACY_NAME VARCHAR2(10) NOT NULL
22 );
23
24 --2) CREATING INTEREST_AREA_TBL
25 CREATE TABLE INTEREST_AREA_TBL (
26     INTEREST_ID NUMBER NOT NULL PRIMARY KEY,
27     INTEREST_NAME VARCHAR2(25) NOT NULL,
28     INTEREST_EXPL VARCHAR2(250)
29 );
30
31 --3) CREATING MAJOR_TBL
32 CREATE TABLE MAJOR_TBL (
33     MAJOR_ID NUMBER NOT NULL PRIMARY KEY,
34     MAJOR_NAME VARCHAR2(30) NOT NULL
35 );
36
37 --4) CREATING DEGREE_TBL
38 CREATE TABLE DEGREE_TBL (
39     DEGREE_ID NUMBER NOT NULL PRIMARY KEY,
40     DEGREE_NAME VARCHAR2(50) NOT NULL
41 );
42
43 --5) CREATING SKILL_TBL
44 CREATE TABLE SKILL_TBL (
45     SKILL_ID NUMBER NOT NULL PRIMARY KEY,
46     SKILL_NAME VARCHAR2(20) NOT NULL
47 );
48
49 --6) CREATING ROLE_TBL
50 CREATE TABLE ROLE_TBL (
51     ROLE_ID NUMBER NOT NULL PRIMARY KEY,
52     ROLE_NAME VARCHAR2(10) NOT NULL
53 );
54
55 --7) CREATING FEEDBACK_TBL
56 CREATE TABLE FEEDBACK_TBL (
57     FEEDBACK_ID NUMBER NOT NULL PRIMARY KEY,
58     STARS_NO NUMBER(1,0) NOT NULL
59 );
60
```

Figure 36 – Resource tables’ codes

Figure 37 shows other tables which are not resource tables anymore. As you can see, the file format is sql and will be run in SQL PLUS Command prompt.

```

61
62 -- 8) CREATING OTHER TABLES:
63 -----
64
65 -- 8) CREATING USER_TBL
66 CREATE TABLE USER_TBL (
67     USER_ID NUMBER NOT NULL PRIMARY KEY,
68     F_NAME VARCHAR2(100) NOT NULL,
69     L_NAME VARCHAR2(100) NOT NULL,
70     U_EMAIL VARCHAR2(100) NOT NULL,
71     PASSWORD VARCHAR2(100) NOT NULL,
72     U_PICTURE BLOB
73 )
74
75
76 -- 9) CREATING JOB_TBL
77 CREATE TABLE JOB_TBL (
78     JOB_ID NUMBER NOT NULL PRIMARY KEY,
79     JOB DATE NOT NULL,
80     PRIVACY_ID NUMBER NOT NULL,
81     USER_ID NUMBER NOT NULL
82 )
83
84 -- 10) CREATING USER_ROLE_TBL
85 CREATE TABLE USER_ROLE_TBL (
86     U_ROLE_ID NUMBER NOT NULL PRIMARY KEY,
87     ROLE_ID NUMBER NOT NULL,
88     USER_ID NUMBER NOT NULL
89 )
90
91 -- 11) CREATING USER_LOCATION_TBL
92 CREATE TABLE USER_LOCATION_TBL (
93     U_LOCATION_ID NUMBER NOT NULL PRIMARY KEY,
94     U_COUNTRY VARCHAR2(100) NOT NULL,
95     U_STATE VARCHAR2(100) NOT NULL,
96     U_CITY VARCHAR2(100) NOT NULL,
97     PRIVACY_ID NUMBER NOT NULL,
98     USER_ID NUMBER NOT NULL
99 )
100
101 -- 12) CREATING WORK_TBL
102 CREATE TABLE WORK_TBL (
103     WORK_ID NUMBER NOT NULL PRIMARY KEY,
104     COMPANY_NAME VARCHAR2(100) NOT NULL,
105     POSITION VARCHAR2(100) NOT NULL,
106     WORK_CITY VARCHAR2(100) NOT NULL,
107     WORK_STATE VARCHAR2(100) NOT NULL,
108     WORK_CITY VARCHAR2(100) NOT NULL,
109     S_DATE DATE,
110     E_DATE DATE,
111     PRIVACY_ID NUMBER NOT NULL,
112     USER_ID NUMBER NOT NULL
113 )
114
115 -- 13) CREATING USER_INTEREST_TBL
116 CREATE TABLE USER_INTEREST_TBL (
117     U_INTEREST_ID NUMBER NOT NULL PRIMARY KEY,
118     INTEREST_ID NUMBER NOT NULL,
119     USER_ID NUMBER NOT NULL
120 )
    
```

Structured Query Language file length: 15098 lines: 545 Ln: 26 Col: 36 Sel: 0 Dos/Windows APMS RMC

Figure 37 - Other SQL tables rather than resource tables

4.5 Oracle Table Relationships

After creating tables, it is the time for relationship structures. This part is very important since it needs double attention and differentiation between one-one, one-many, and many-many relationships.

To summarize the general rules that I used, I should mention that one-one is very easy to understand. In this relationship each of the two tables can have only one data in another table. For example, imagine USER_TBL and DOB_TBL. One user can only have one date of birth and one date of birth only belongs to one user. In this case, USER_TBL is our main table because a user needs to first register her information in that table for starting to be able to use the website. So I have created the user table first and then the DOB_TBL. Regarding that, the foreign keys should go in the secondary table; the one that has less priority for user initial actions. The alteration code for DOB_TBL is shown in Figure 38.

```
--DOB_TBL FKS  
  
ALTER TABLE DOB_TBL  
ADD CONSTRAINT PRIVACY_ID FOREIGN KEY (PRIVACY_ID)  
REFERENCES PRIVACY_TBL (PRIVACY_ID);  
  
ALTER TABLE DOB_TBL  
ADD CONSTRAINT USER_ID FOREIGN KEY (USER_ID)  
REFERENCES USER_TBL (USER_ID);
```

Figure 38 - Alteration code for creating relationship between DOB_TBL and USER_TBL

One-many relationship could be explained in users and their work experiences. In this example, each user could have had several jobs; and many job experiences can belong to one user. So, WORK_TBL and USER_TBL has this relationship, and again, because the USER_TBL is needed first by any user, it is considered main, and WORK_TBL is considered secondary. Figure 39 shows how the alteration for WORK_TBL adds constrains.

```
--WORK_TBL FKS  
  
ALTER TABLE WORK_TBL  
  
ADD CONSTRAINT PRIVACY_FK_W FOREIGN KEY (PRIVACY_ID)  
  
REFERENCES PRIVACY_TBL (PRIVACY_ID);  
  
  
  
ALTER TABLE WORK_TBL  
  
ADD CONSTRAINT USER_FK_W FOREIGN KEY (USER_ID)  
  
REFERENCES USER_TBL (USER_ID);
```

Figure 39 – Work table alteration by adding foreign keys

Last but not least, there is many-many relationship. This is a situation where for optimizing the tables and preventing data redundancy, we need to add a relationship table in between. An example that can clearly show this case is ROLE_TBL and USER_TBL. Each user can have many roles and one role can be assigned to many users. Thereby, the best practice is to create a USER_ROLE_TBL to relate these two tables while prevent redundancy of some data like names and profile pictures, which could save a lot of space. Figure 40 shows the codes of the relation table and its relation to USER_TBL and ROLE_TBL.

Appendix E shows all the SQL/Plus codes used in the database side.

```
--USER_ROLE_TBL FKS  
  
ALTER TABLE USER_ROLE_TBL  
  
ADD CONSTRAINT ROLE_ID FOREIGN KEY (ROLE_ID)  
  
REFERENCES ROLE_TBL (ROLE_ID);  
  
  
ALTER TABLE USER_ROLE_TBL  
  
ADD CONSTRAINT USER_FK_UR FOREIGN KEY (USER_ID)  
  
REFERENCES USER_TBL (USER_ID);
```

Figure 40 – Relational table for USER_TBL and ROLE_TBL

4.6 PHP/Oracle coding

The part that actually make the whole pages work is when SQL queries are placed in PHP codes and the front end gets connected to the back end. Appendix D includes all the codes. The basic is that we create general pages like home.php. They all include the template files. Figure 35 showed one of the pages which contained the PHP/Oracle functions.

Chapter 5: Testing Phase

For testing one of the examples, consider one of the users who is the admin of the website. The user's information exists in USER_TBL, so, she can log in. In the header.php, the function "is_admin ('user_id')" checks the USER_ROLE_TBL to see if the user is admin or not. If yes, the header.php will contain the admin.php, else it won't. In our example, the user is Fatemeh. She is the admin. Figure 41 shows the header. For admin convenience, the left sidebar is considered as a part of her page, so that she can use the website like other users.

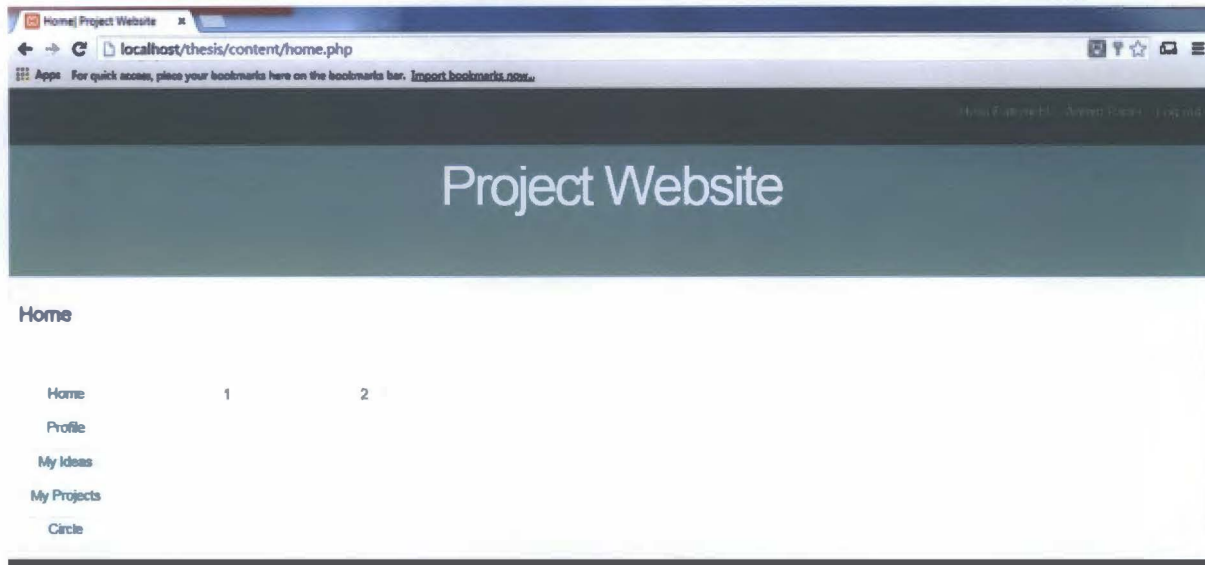


Figure 41 – Admin.php is read from the header.php since the user is admin

The default menu is Home.php. In this page all the projects which are either public, or shared with the user, or owned by user. As you can see in Figure 41, the home.php returns two project ids for the user. However, the user is participating only in one project.

Figure 42 shows “My project” which contains one project.

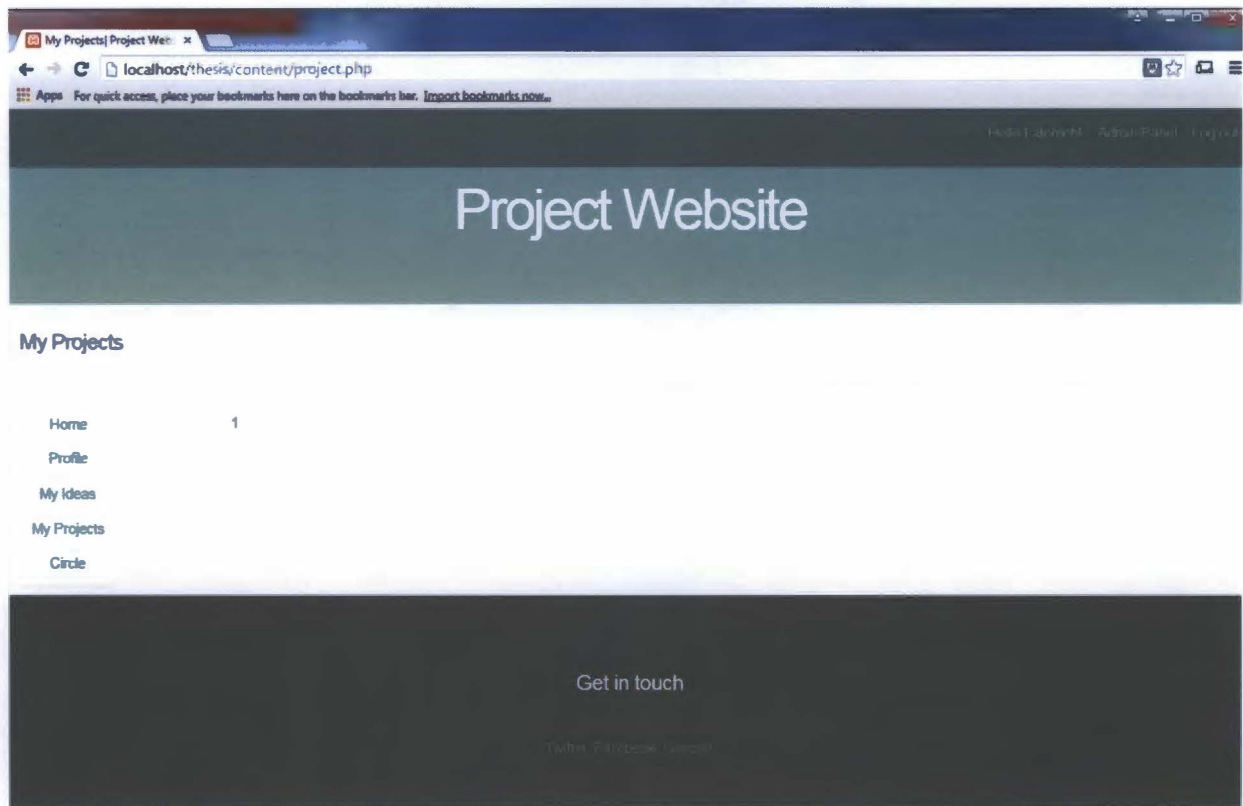


Figure 42 – My project page

Admin panel has not been programmed in this project thesis. But the admin can see other users in this page and access their data to some extent. She can edit some of the fields and activate/deactivate any user.

The circle page is the one which shows the people who are followed by user and the people who are following the user. In this page, a function named “count_circle(\$session_id)” counts the number of people in the whole circle. Figure 43 shows a screen shot of this page.

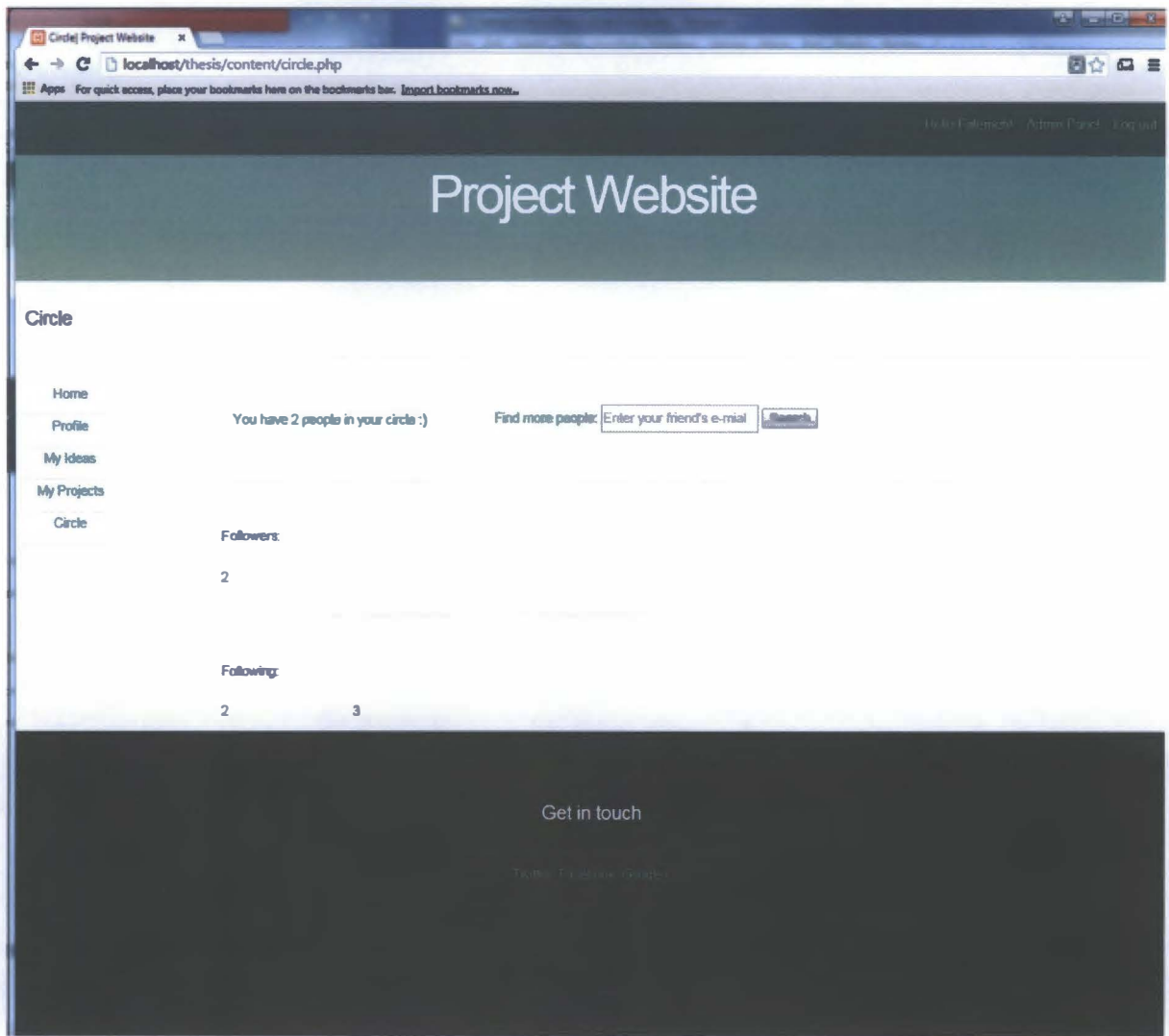


Figure 43 – Circle page screen shot

Chapter 6: Conclusion

The thesis proposed a project portal to solve one of the observed problems in research groups by helping researchers find each other easier, share their information to the extend they need to, and start doing projects. All of these actions are less time-consuming and it will develop the research skills.

After doing some research about different kinds of programming languages, like Java, PHP, JavaScript, Python, and Objective-C, it is concluded that PHP programming language is better for implementation. On the other hand, the need for a database to save all the users' records in the system was deniable. Among three most popular databases, Oracle, MySQL, and Microsoft SQL Server, Oracle was chosen. The project is implemented by Oracle 11 g database technology with PHP in XAMPP Server v3.2.1.

The tables' structure and design were an important part of this project. With the help of flowcharts, the planning was made easier. The flowcharts led planning to tables and the tables led the planning to sample tables. After designing completion, the depicted tables could be used as the road map for the whole projects.

The initial idea of this thesis is based on collaboration and team working. The result of this work could be used in a bigger scale. For example, instead of limiting its application to EIU students, it can be applied in other universities, or even businesses.

6.1 Future Work

The work which is done in this thesis is more focused on database designing part and the project orientation functions and operations. From graphic design side, the project does not need more perfections. However, it can be developed from PHP/Oracle point of view. Some of the possible future works are as follows:

- Profile page can show more information of the user since it already has the user session id. Plus the fact that the data can be changed by the logged in user in her/his profile.
- In the circle page, the data of the followers and following users can be retrieved from the database easily because with help of `get_data($user_id)` function the user ids are accessible. This simply means that other kinds of data from the users are accessible too.
- The idea page is similar to project page. The functions can be easily made similar to the project retrieving data functions.
- The home page can contain more information from the content.
- Although the graphic part does not need anything to be added at this moment, in the future, there might be more complex JavaScript added to each page.

References

- Alex. 10 best programming languages of 2015 you should know. (2015). Retrieved from <https://www.devsaran.com/blog/10-best-programming-languages-2015-you-should-know>
- Apache Friends. (2015). Download [Date file]. Retrieved from <https://www.apachefriends.org/download.html>
- Cass, S. The 2015 top ten programming languages. (2015). *Spectrum.ieee*. Retrieved from <http://spectrum.ieee.org/computing/software/the-2015-top-ten-programming-languages>
- Codecourse. (2012, May 11). PHP tutorials: Register & login [Part 4]. Database connection. Retrieved from <https://www.youtube.com/watch?v=0ol6C0ETQqM>
- Codecourse. (2012, May 11). PHP tutorials: Register & login [Part 5]. User login (part 1). Retrieved from <https://www.youtube.com/watch?v=ueruxlD0Smo>
- Codecourse. (2012, May 11). PHP tutorials: Register & login [Part 5]. User login (part 2). Retrieved from <https://www.youtube.com/watch?v=Til3oVNIho4>
- Codecourse. (2012, May 11). PHP tutorials: Register & login [Part 7]. User data (part 1). Retrieved from <https://www.youtube.com/watch?v=JhLnS0mXr0M>
- Codecourse. (2012, May 11). PHP tutorials: Register & login [Part 7]. User data (part 2). Retrieved from <https://www.youtube.com/watch?v=yX1Eushqclw>
- College Career Life. Six main career interest areas. (2008). Retrieved from <http://collegecareerlife.net/interest-areas-choose-a-career/>
- Duet, D. (2015, June 11). Retrieved from <https://www.youtube.com/user/GoldmanSachs?v=h-mvgkRWP34>

Examples of skills. In *Examples Your Dictionary*. Retrieved from

<http://examples.yourdictionary.com/examples-of-skills.html>

Gokak, s. (2009, August 26). How to visualize the relationship of tables in an Oracle

database. [Answer 1]. Answer posted to

<http://stackoverflow.com/questions/1333400/how-to-visualize-the-relationship-of-tables-in-an-oracle-database>

Hendricks, D. (2015). How technology has revitalized project management. *Tech.co*.

Retrieved from <http://tech.co/technology-revitalized-project-management-2015-03>

How to start programming. (2015). *PHP*. Retrieved from

<http://howtostartprogramming.com/PHP/>

Top 10 databases you should learn in 2015. (2015). *I'm programmer*. Retrieved from

<http://www.improgrammer.net/top-10-databases-should-learn-2015/>

Liquid Planner. (2015). Features. Retrieved from <http://www.liquidplanner.com/features/>

Martin. (2015). Top programming languages used in web development. *Entrepreneurial-insights*. Retrieved from <http://www.entrepreneurial-insights.com/programming-languages-web-development>

Morris, J. (2013, April 28). How to Create a Website Template Using PHP. Retrieved from

<https://www.youtube.com/watch?v=lUFXLHqi-uQ>

My VMware. (2015). Desktop & end-user computing. Retrieved from

<https://my.vmware.com/web/vmware/downloads>

Open source. (n. d.). In *Webopedia*. Retrieved from

http://www.webopedia.com/TERM/O/open_source.html

- Oracle. (2015). Getting started with Oracle SQL Developer [Data file]. Retrieved from http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/11g/r2/prod/appdev/sqldev/sqldev_mngdb/sqldev_mngdb_otn.htm
- Oracle. (2014). Oracle holds #1 DBMS market share worldwide for 2013 [Data set]. Retrieved from <http://www.oracle.com/us/corporate/features/number-one-database/index.html>
- Project Insight. 5 basic phases of project management (Web publication). Retrieved from <http://www.projectinsight.net/project-management-basics/basic-project-management-phases>
- Purdue Online Writing Lab. (2015). Reference list: electronic sources (Web publications)[Data set]. Retrieve from <https://owl.english.purdue.edu/owl/resource/560/10/>
- Sharma, M. (2014, January 11). Oracle Database tutorials 1: How to install Oracle Database 11g on windows 7. Retrieved from <https://www.youtube.com/watch?v=0NLsJQCvKXY>
- Study. List of degrees: most popular degree programs by level. (n. d.). Retrieved from http://study.com/list_of_degrees.html
- The Art of Web. (2014). HTML: HTML5 form validation examples. Retrieved from <http://www.the-art-of-web.com/html/html5-form-validation/>
- TVAC Studio. (2013, February 23). PHP tutorial [Part 2]. Admin panel system. <https://www.youtube.com/watch?v=1UIGY1mNrCY>
- Web Technology Surveys. (2015). Usage statistics and market share of PHP for websites [Data set]. Retrieved from <http://w3techs.com/technologies/details/pl-php/all/all>

Appendix A: CSS and JS codes

A-The CSS code

```
/*  
This style is designed by Fatemeh  
for thesis purpose only!  
This is the template used in the first page!  
*/
```

```
@charset 'UTF-8';
```

```
/*  
*****  
/* Basic */  
*****  
*/
```

```
body  
{  
margin : 0;  
padding :0;  
background-color : #161616;  
}
```

```
body,input,textarea,select  
{  
font-family: 'Roboto', sans-serif;
```

```
        font-size: 11pt;
        font-weight: 300;
        line-height: 1.75em;
        color: #777;
    }
```

h1, h2, h3, h4, h5, h6

```
{
font-weight : 300;
color : #777;
}
```

h2

```
{
        letter-spacing: -0.025em;
    }
```

h1 a, h2 a, h3 a, h4 a, h5 a, h6 a

```
{
        color: inherit;
        text-decoration: none;
    }
```

strong, b

```
{
```

```
        font-weight: 500;
        color: #000000;
    }

    em, i
    {
        font-style: italic;
    }

    a
    {
        text-decoration: none;
        color: #80BA8E;
    }

    a:hover
    {
        text-decoration: underline;
    }

    table
    {
        top:0;
        padding: 10px;
        color: #80BA8E;
```

```
}
```

```
sub
```

```
{
```

```
    position: relative;
```

```
    top: 0.5em;
```

```
    font-size: 0.8em;
```

```
}
```

```
sup
```

```
{
```

```
    position: relative;
```

```
    top: -0.5em;
```

```
    font-size: 0.8em;
```

```
}
```

```
hr
```

```
{
```

```
    border: 0;
```

```
    border-top: solid 1px #ddd;
```

```
    margin: 2em 0 0 0;
```

```
    padding: 2em 0 0 0;
```

```
}
```

```
/* Sections/Articles */
```

```
    section,  
    article  
  {  
    margin-bottom: 3em;  
  }  
  
  section > :last-child,  
  article > :last-child  
  {  
    margin-bottom: 0;  
  }  
  
  section:last-child,  
  article:last-child  
  {  
    margin-bottom: 0;  
  }  
  
  .row > section,  
  .row > article  
  {  
    margin-bottom: 0;  
  }  
  
  /* Grid */
```

/* Cells */

```
.\31 2u { width: 100% }
.\31 1u { width: 91.6666666667% }
.\31 0u { width: 83.3333333333% }
.\39 u { width: 75% }
.\38 u { width: 66.6666666667% }
.\37 u { width: 58.3333333333% }
.\36 u { width: 50% }
.\35 u { width: 41.6666666667% }
.\34 u { width: 33.3333333333% }
.\33 u { width: 25% }
.\32 u { width: 16.6666666667% }
.\31 u { width: 8.3333333333% }
.\-11u { margin-left: 91.6666666667% }
.\-10u { margin-left: 83.3333333333% }
.\-9u { margin-left: 75% }
.\-8u { margin-left: 66.6666666667% }
.\-7u { margin-left: 58.3333333333% }
.\-6u { margin-left: 50% }
.\-5u { margin-left: 41.6666666667% }
.\-4u { margin-left: 33.3333333333% }
.\-3u { margin-left: 25% }
.\-2u { margin-left: 16.6666666667% }
.\-1u { margin-left: 8.3333333333% }
```



```
.row > * {  
    padding: 50px 0 0 50px;  
    float: left;  
    -moz-box-sizing: border-box;  
    -webkit-box-sizing: border-box;  
    -o-box-sizing: border-box;  
    -ms-box-sizing: border-box;  
    box-sizing: border-box;  
}
```

```
.row + .row > * {  
    padding-top: 50px;  
}
```

```
.row {  
    margin-left: -50px;  
}
```

```
/* Rows */
```

```
.row:after {  
    content: " ";  
    display: block;
```

```
        clear: both;

        height: 0;
    }

    .row:first-child > * {
        padding-top: 0;
    }

    .row > * {
        padding-top: 0;
    }

    /* Modifiers */

    /* Flush */

    .row.flush {
        margin-left: 0;
    }

    .row.flush > * {
        padding: 0 !important;
    }

    /* Quarter */
```

```
.row.quarter > * {  
    padding: 12.5px 0 0 12.5px;  
}  
  
.row.quarter + .row.quarter > * {  
    padding-top: 12.5px;  
}  
  
.row.quarter {  
    margin-left: -12.5px;  
}  
  
/* Half */  
  
.row.half > * {  
    padding: 25px 0 0 25px;  
}  
  
.row.half + .row.half > * {  
    padding-top: 25px;  
}  
  
.row.half {  
    margin-left: -25px;
```

```
}
```

```
/* One and (a) Half */
```

```
.row.oneandhalf > * {  
    padding: 75px 0 0 75px;
```

```
}
```

```
.row.oneandhalf + .row.oneandhalf > * {  
    padding-top: 75px;
```

```
}
```

```
.row.oneandhalf {  
    margin-left: -75px;
```

```
}
```

```
/* Double */
```

```
.row.double > * {  
    padding: 100px 0 0 100px;
```

```
}
```

```
.row.double + .row.double > * {  
    padding-top: 100px;
```

```
}
```

```
        .row.double {
            margin-left: -100px;
        }

/*****
/* Header */
*****/

#header
{
    height: 550px;

    position: relative;

    background: url(../media/header.jpg) no-repeat bottom center;
    background-attachment: fixed;
    background-size: cover;
    text-align: center;

}

/*****
/* Logo */
*****/

#logo
{
```

```
}

```

```
#logo a

```

```
{

```

```
    text-decoration: none;

```

```
    font-weight: 300;

```

```
    line-height: 1em;

```

```
    font-size: 7em;

```

```
    color: #FFF;

```

```
    letter-spacing: -0.05em;

```

```
    margin: 0 0 0.125em 0;

```

```
}

```

```
#logo span

```

```
{

```

```
    font-size: 1.4em;

```

```
    color: #FFF;

```

```
}

```

```
/* ***** */

```

```
/* Nav */

```

```
/* ***** */

```

```
#nav
```

```
{
```

```
}
```

```
#nav-wrapper
```

```
{
```

```
    background: rgba(0,0,0,.1);
```

```
    position: absolute;
```

```
    top: 0;
```

```
    left: 0;
```

```
    width: 100%;
```

```
    height: 100px;
```

```
}
```

```
/******  
/* Footer */  
/******
```

```
/* Footer */
```

```
/******
```

```
#footer
```

```
{
```

```
    position: relative;
```

```
        text-align: center;
        padding:60px;
    }

    #footer header h2
    {
        color: #FFF !important;
    }

    #footer header .byline
    {
        color: rgba(255,255,255,.2);
    }

    /*****
    /* Copyright                               */
    *****/

    #copyright
    {
        position: relative;
        text-align: center;
        color: #774535;
    }
```



```
#copyright .container
{
    padding: 3em 0em;
    border-top: 1px solid;
    border-color: rgba(255,255,255,.05);
    color: rgba(255,255,255,.1);
}

#copyright a
{
    text-decoration: none;
    color: rgba(255,255,255,.2);
}

/*****
/* Featured */
*****/

#featured
{
    padding: 10px;
    position: relative;
    background: #f2f2f2;
    text-align: center;
}
```

```
#featured h3
{
    display: block;
    font-weight: 300;
}
```

A-JS Codes

```
function confirmEmail() {
    var email = document.getElementById("email").value;
    var confemail = document.getElementById("confemail").value;
    if(email != confemail) {
        alert('Email Not Matching!');
        document.getElementById("confemail").value="";
    }
}

function checkForm(form)
{
    re = /^w+$/;
    if(!re.test(form.fname.value)) {
        alert("Error: First name must contain only letters!");
        form.fname.focus();
        return false;
    }
}

re = /^w+$/;
```

```
if(!re.test(form.lname.value)) {
    alert("Error: Last name must contain only letters!");
    form.lname.focus();
    return false;
}

if(form.pass.value == form.email.value) {
    alert("Error: Password must be different from Username!");
    form.pass.focus();
    return false;
}

re = /[0-9]/;

if(!re.test(form.pass.value)) {
    alert("Error: password must contain at least one number (0-9)!");
    form.pass.focus();
    return false;
}

re = /[a-z]/;

if(!re.test(form.pass.value)) {
    alert("Error: password must contain at least one lowercase letter (a-z)!");
    form.pass.focus();
    return false;
}

re = /[A-Z]/;

if(!re.test(form.pass.value)) {
    alert("Error: password must contain at least one uppercase letter (A-Z)!");
```

```
    form.pass.focus();  
    return false;  
}  
  
alert("You entered a valid password: " + form.pwd1.value);  
return true;  
}
```

Appendix B: PHP Template Codes

B-Function.php

```
<?php
include('../template/class-them-method.php');

function do_main_nav(){
    global $dtm;

    $class = "main_nav";

    $items_array = array (
        array('text' => 'home', 'url' => '/web-temp'),
        array('text' => 'About', 'url' => 'about.php')
    );

    return $dtm->navigation($items_array, $class);
}

function do_html_title($page_title){
    $title= $page_title . '| Project Website';

    return $title;
}

function home_content(){

}

?>
```

B-Header.php

```
<?php
include('../template/function.php');
?>
<!DOCTYPE html>
<html>
  <head>
    <title><?php echo do_html_title($the_title);?></title>
    <link rel="stylesheet" type="text/css" href="../css/t_style.css" />
  </head>
  <body>
    <div id="header">
      <div id="nav-wrapper">
        <!-- Nav -->
        <nav id="nav">
          <table>
            <tr>
              <td>Hello <?php
                if(logged_in()===true){
                  $r= get_data($session_id);
                  print_r($r['F_NAME'][0]);
                } else{
```



```
<a href="#">Project Website</a></br>
```

```
</div>
```

```
</div>
```

```
</div>
```

B-Footer.php

```
<div id="footer">
```

```
<div>
```

```
<section>
```

```
<header>
```

```
<h2>Get in touch</h2>
```

```
<span class="byline">Follow us to get informed
```

```
about the last updates</span>
```

```
</header>
```

```
<div>
```

```
<a href="#">Twitter</a>&nbsp;&nbsp;&nbsp;<a
```

```
href="#">Facebook</a>&nbsp;&nbsp;&nbsp;<a href="#">Google+</a>
```

```
</div>
```

```
</section>
```

```
</div>
```

```
</div>
```

```
<!-- Copyright -->
```



```
<div id="copyright">
    <div class="container">
        Design: <a href="mailto:fkhastkhodaard@eiu.edu">Fatemeh</a>
    </div>
</div>

</body>
</html>
```

B-Sidebar.php

```
<div>
    <ul>
        <li class="active"><a href=" ../content/home.php">Home</a></li>
        <li><a href=" ../content/profile.php">Profile</a></li>
        <li><a href=" ../content/idea.php">My Ideas</a></li>
        <li><a href=" ../content/project.php">My Projects</a></li>
        <li><a href=" ../content/circle.php">Circle</a></li>
    </ul>
</div>
```

B-Class-them-method.php

```
<?php
if (!class_exists('ProjectWebsiteMethods')){
    class ProjectWebsiteMethods {
```

```
public function navigation($items_array, $class){
    $nav='<ul class="'. $class. ">';

    foreach ($items_array as $item){
        $nav .= '<li><a href="'. $item['url']. ">'. $item['text'] . '</a></li>';
    }

    $nav .= '</ul>';

    return $nav;
}
}
}

$dtm = new ProjectWebsiteMethods;
?>
```

Appendix C: PHP/Oracle Functions

```
<?php
function db_connect()
{
    $db = "localhost/orcl.16.173.130";

    if ($c=oci_connect("project", "project", $db)){
        return $c;
    } else {
        $err = OCIError();
        echo "Oracle Connect Error " . $err[text];
    }
}

function user_exists($username){
    $con2 = db_connect();
    $username = sanitize_email ($username);
    $query= "select u_email from user_tbl where u_email = '$username'";
    $stid = OCIparse($con2, $query);
    OCIexecute($stid);
    oci_fetch_all($stid, $res);
    //var_dump($res);
    if (OCIRowCount($stid) != $username)
```

```
{ return true;} else
```

```
{ return false;}
```

```
}
```

```
function user_active($username){
```

```
$con2 = db_connect();
```

```
$username = sanitize_email ($username);
```

```
$query= "select active from user_tbl where u_email = '$username'";
```

```
$s = oci_parse($con2, $query);
```

```
oci_execute($s);
```

```
oci_fetch_all($s, $res);
```

```
//var_dump($res);
```

```
if ($res["ACTIVE"][0] == 1) {
```

```
return true;
```

```
}
```

```
else {
```

```
return false;}
```

```
}
```

```
//=====create session &&
```

```
login=====
```

```
function get_id($username){
```

```
$con2 = db_connect();
```

```
$username=sanitize_email($username);
```

```
$query = "select user_id from user_tbl where u_email = '$username'";  
$s = oci_parse($con2, $query);  
oci_execute($s);  
oci_fetch_all($s, $res);  
return $res["USER_ID"][0];  
}
```

```
function login($username,$password){  
    $con2 = db_connect();  
    $username = sanitize_email ($username);  
    $query="select password from user_tbl where u_email='$username'";  
    $result=oci_parse($con2, $query);  
    oci_execute($result);  
    oci_fetch_all($result, $res);  
    $id=get_id($username);  
    //var_dump($res);  
    if( $res["PASSWORD"][0] == $password) { return $id; }  
    else{ return false;}  
}  
  
//=====Sign up  
functions=====
```

```
function next_user_id(){  
    $con2 = db_connect();
```

```
$query="select max(user_id) from user_tbl";

$result=oci_parse($con2, $query);

oci_execute($result);

oci_fetch_all($result, $res);

$r= $res["MAX(USER_ID)"][0];

return $r+1;

}

function insert_user($fname, $lname, $email, $password){

$con2 = db_connect();

$next = next_user_id();

$query="insert into user_tbl (user_id, f_name, l_name, u_email, password, u_picture,
active)

values ( '$next', '$fname', '$lname', '$email', '$password', '', 0 )";

$result=oci_parse($con2, $query);

oci_execute($result);

if($result){

return true;

} else {return false;}

}

function send_activation_email($email){
```

```
$message = 'Thank you for registration. For activating your account, you need to click on  
this: http://localhost/thesis/activated.php';
```

```
//make the message a little in shape
```

```
$message = wordwrap($message, 70, "\r\n");
```

```
mail($email, 'Activation Email', $message);
```

```
}
```

```
function activate_user($email){
```

```
    $con2 = db_connect();
```

```
    $query="update user_tbl set active=1 where u_email='$email'";
```

```
    $result=oci_parse($con2, $query);
```

```
}
```

```
//=====After
```

```
login=====
```

```
function logged_in(){
```

```
    if(isset($_SESSION['user_id'])){
```

```
        return true; }
```

```
    else{
```

```
        return false;
```

```
    }
```

```
}
```

```
function get_data($session_id){
    $con2 = db_connect();
    $query= "select * from user_tbl where user_id = '$session_id'";
    $data= oci_parse($con2, $query);
    oci_execute($data);
    oci_fetch_all($data, $res);
    return($res);
}

function is_admin($login){
    $con2 = db_connect();
    $query= "select role_id from user_role_tbl where user_id = '$login'";
    $r= oci_parse($con2, $query);
    oci_execute($r);
    oci_fetch_all($r, $res);
    $result= $res["ROLE_ID"][0];
    if($result == 4){
        return true;
    } else {
        return false;
    }
}
```



```
}
```

```
//-----Project-----
```

```
-----
```

```
//returns private projects to the owners for their editing
```

```
function get_private_project($session_id){
```

```
    $con2 = db_connect();
```

```
    $query= "select project_id from user_project_tbl where user_id='$session_id' &&
```

```
    privacy_id=2 && role_id=3";
```

```
    $data= oci_parse($con2, $query);
```

```
    oci_execute($data);
```

```
    oci_fetch_all($data, $res);
```

```
    return $res["PROJECT_ID"];
```

```
}
```

```
//get the id of projects that the user is in them
```

```
function get_projects($session_id){
```

```
    $con2 = db_connect();
```

```
    $query= "select project_id from user_project_tbl where user_id = '$session_id'";
```

```
    $data= oci_parse($con2, $query);
```

```
    oci_execute($data);
```

```
    oci_fetch_all($data, $res);
```

```
    if(!empty($res["PROJECT_ID"])){
```

```
return($res);

} else

{ return false;

}

}

//team privacy

function get_team_privacy_project($session_id){

$con2 = db_connect();

$query= "select project_id from user_project_tbl where user_id='$session_id' &&

privacy_id=5 && (role_id=1 or role_id=2)";

$data= oci_parse($con2, $query);

oci_execute($data);

oci_fetch_all($data, $res);

return $res["PROJECT_ID"];

}

//-----Ideas-----

-----

//get the ideas which the user has registered

function get_ideas($session_id){

$con2 = db_connect();

$query= "select idea_id from user_idea_tbl where user_id = '$session_id'";
```

```
$data= oci_parse($con2, $query);
oci_execute($data);
oci_fetch_all($data, $res);
if(!empty($res["IDEA_ID"])){
return($res);
} else
{ echo 'Currently, there is no registered idea of yours, but you can always share what you
have in mind with people you want! <br>';
}
}
```

```
//-----home-----
```

```
-----
```

```
function get_public_projects(){
$con2 = db_connect();
$query= "select project_id from user_project_tbl where privacy_id = 2";
$data= oci_parse($con2, $query);
oci_execute($data);
oci_fetch_all($data, $res);
return $res["PROJECT_ID"];
}
```

```
function home_project_widget($session_id){
$a=get_projects($session_id);
$b=get_public_projects();
if(empty($a)){
return $b;
}elseif(empty($b)){
return $a;
} else{
$c=array_intersect_key($a, $b);
$d=array_diff($b, $c);
$r= array_merge($a , $d);
//$r=array_unique($r);
return $r["PROJECT_ID"];
}
}

//-----circle-----
-----

//this function returns the people who are following the user
function follower($session_id){
$con2 = db_connect();
$query= "select follower_id from following_tbl where followed_id = '$session_id'";
$data= oci_parse($con2, $query);
```

```
oci_execute($data);

oci_fetch_all($data, $res);

if(!empty($res["FOLLOWER_ID"])){
return($res["FOLLOWER_ID"]);
} else
{ return 0;
}
}

//this function returns the people who are followed by user as the follower
function followed($session_id){
$con2 = db_connect();

$query= "select followed_id from following_tbl where follower_id = '$session_id'";

$data= oci_parse($con2, $query);

oci_execute($data);

oci_fetch_all($data, $res);

if(!empty($res["FOLLOWED_ID"])){
return($res["FOLLOWED_ID"]);
} else
{ return 0;
}
}
```

```
//count the number of people in the user's circle and return that number
```

```
function count_circle($session_id){  
    $r= followed($session_id);  
    $s= follower($session_id);  
    $a=array_intersect_key($r, $s); //returns the common contents  
    $b= array_diff($s, $a); //  
    $c= array_merge($b, $r);  
    return count($c);  
}
```

```
?>
```

Appendix D: PHP/Oracle Files

D-Home.php

```
<?php
$the_title = 'Home';
include('../core/init.php');
include('../template/header.php');

?>

<div id="featured">
    <header><h2>Home</h2></header>
    <hr />
    <div class="row">
        <section class="2u" id="sidebar">
            <?php include('../template/sidebar.php');
            ?>
        </section>
        <section class="8u">
            <div class="row">
                <?php
                    $r=home_project_widget($session_id);

                    $x= count($r);
```

```
//          for($i=0;$i<$x;$i++){
            foreach ($r as $row){

                echo '<section class="2u">';
                echo '<a href="#">'.$row.'</a>';
//          echo '<a href="#">'.$row[$i].</a>';
                echo '</section>';}

            }
        ?>
    </div>

    </section>
</div>

</div>

<?php include('../template/footer.php'); ?>
```


D-Profile.php

```
<?php
$the_title = 'Profile';
include('../core/init.php');
include('../template/header.php');

?>

<div id="featured">

    <header><h2>Profile</h2></header>

    <hr />

    <div class="row">

        <section class="2u" id="sidebar">

            <?php include('../template/sidebar.php');

            ?>

        </section>

        <section class="10u">

            <div class="row">

                <section class="4u">

                    <div id="profile_picture">

                        <?php

                            $a=get_data($session_id);
```

```
print_r($a['U_PICTURE'][0]);
```

```
?>
```

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

```
<section class="4u" id="center">
```

```
<?php
```

```
print_r($a['F_NAME'][0]);
```

```
echo ' ';
```

```
print_r($a['L_NAME'][0]);
```

```
?>
```

```
<section>
```

```
</div>
```

```
</section>
```

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

```
<section class="6u">
```

```
<form action="upload.php" method="post" enctype="multipart/form-  
data">
```

Select image to upload:

```
<input type="file" name="fileToUpload" id="fileToUpload">
```

```
<input type="submit" value="Upload Image" name="submit">
</form>
</section>
</div>
    </div>

</div>

<?php include('../template/footer.php'); ?>
```

D-Project.php

```
<?php
$the_title = 'My Projects';
include('../core/init.php');
include('../template/header.php');
?>

<div id="featured">

    <header><h2>My Projects</h2></header>

    <hr />

    <div class="row">

        <section class="2u" id="sidebar">
```

```
<?php include('../template/sidebar.php');
?>

</section>

<section class="8u">

<?php

if(!empty(get_projects($session_id))){
$r=get_projects($session_id);
$x= count($r);
for($i=0;$i<$x;$i++){
foreach ($r as $row){
echo '<section class="2u">';
echo '<a href="#">' . $row[$i] . '</a>';
echo '</section>';}
}
} else {echo 'Currently, there is no registered project for you, but you can always start what
you are passionate about :) <br>';}

?>

</section>

</div>
```

```
</div>
```

```
<?php include('../template/footer.php'); ?>
```

D-Idea.php

```
<?php
```

```
$the_title = 'My Ideas';
```

```
include('../core/init.php');
```

```
include('../template/header.php');
```

```
?>
```

```
<div id="featured">
```

```
<header><h2>My Ideas</h2></header>
```

```
<hr />
```

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

```
<section class="2u" id="sidebar">
```

```
<?php include('../template/sidebar.php');
```

```
?>
```

```
</section>
```

```
<section>
```

```
<?php
```

```
//get_ideas($session_id);
```

```
?>
```

```
</section>
```

```
</div>
```

```
</div>
```

```
<?php include('../template/footer.php'); ?>
```

D-Circle.php

```
<?php
```

```
$the_title = 'Circle';
```

```
include('../core/init.php');
```

```
include('../template/header.php');
```

```
?>
```

```
<div id="featured">
```

```
<header><h2>Circle</h2></header>
```



```
$x= count($r);
```

```
foreach ($r as $row){  
    echo '<section class="2u">';  
    echo '<a href="#">' . $row . '</a>';  
    echo '</section>';}
```

```
?>
```

```
</div>
```

```
<hr />
```

```
<h4> Following:</h4>
```

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

```
<?php
```

```
$r=followed($session_id);
```

```
$x= count($r);
```

```
foreach ($r as $row){  
    echo '<section class="2u">';
```



```
        echo '<a href="#">' . $row . '</a>';  
        echo '</section>';  
    }  
  
?>  
  
</div>  
  
        </section>  
    </div>  
  
</div>
```

```
<?php include('../template/footer.php'); ?>
```

D-Admin.php

```
<?php  
$the_title = 'Admin Panel';  
include('../core/init.php');  
include('../template/header.php');  
  
?>  
  
<div id="featured">
```

```
<header><h2>Admin Panel</h2></header>

    <hr />

    <div class="row">

        <section class="2u" id="sidebar">

            <?php include('../template/sidebar.php');

            ?>

        </section>

        <section>

            <?php

                echo 'admin panel!!!'

            ?>

        </section>

    </div>

</div>

<?php include('../template/footer.php'); ?>
```

D-Search.php

```
<?php
```

```
include '../core/init.php';

if(isset($_GET)===true){

$email = $_GET['email'];

$id= get_id($email);

if ($id === $session_id){

echo "That's yourself! :)";

} elseif(user_exists($email)===true){

$r=get_data($id);

?>

<a href="#"> <?php print_r($r['F_NAME'][0]); echo ' '; print_r($r['L_NAME'][0]); ?> </a>

<?php

} else { echo 'There is not such an email in our database!;}

}

?>
```

D-Logout.php

```
<?php

session_start();

session_destroy();

header('location: ../index.php');

?>
```

Appendix E: SQL/Plus queries

E-Creating Tables and Relationships

--logging in to the privileged user of the Oracle db "SYSTEM"

CONNECT system/&system_password;

--creating PROJECT user

--DROP USER PROJECT CASCADE;

CREATE USER project IDENTIFIED BY "project";

GRANT ALL PRIVILEGES TO PROJECT;

--logging in to the new user "PROJECT"

CONNECT PROJECT / "project"

--creating tables and their relationship according to the designing phase

-- A) CREATING SOURCE TABLES:

--1) CREATING PRIVACY_TBL

CREATE TABLE PRIVACY_TBL (

 PRIVACY_ID NUMBER NOT NULL PRIMARY KEY,

```
    PRIVACY_NAME VARCHAR2(10) NOT NULL  
);
```

```
--2)CREATING INTEREST_AREA_TBL
```

```
CREATE TABLE INTEREST_AREA_TBL (  
    INTEREST_ID NUMBER NOT NULL PRIMARY KEY,  
    INTEREST_NAME VARCHAR2(25) NOT NULL,  
    INTEREST_EXPL VARCHAR2(250)  
);
```

```
--3)CREATING MAJOR_TBL
```

```
CREATE TABLE MAJOR_TBL (  
    MAJOR_ID NUMBER NOT NULL PRIMARY KEY,  
    MAJOR_NAME VARCHAR2(30) NOT NULL  
);
```

```
--4)CREATING DEGREE_TBL
```

```
CREATE TABLE DEGREE_TBL (  
    DEGREE_ID NUMBER NOT NULL PRIMARY KEY,  
    DEGREE_NAME VARCHAR2(50) NOT NULL  
);
```

```
--5)CREATING SKILL_TBL
```

```
CREATE TABLE SKILL_TBL (  
    SKILL_ID NUMBER NOT NULL PRIMARY KEY,  
    SKILL_NAME VARCHAR2(20) NOT NULL  
);
```

--6) CREATING ROLE_TBL

```
CREATE TABLE ROLE_TBL (  
    ROLE_ID NUMBER NOT NULL PRIMARY KEY,  
    ROLE_NAME VARCHAR2(10) NOT NULL  
);
```

--7) CREATING FEEDBACK_TBL

```
CREATE TABLE FEEDBACK_TBL (  
    FEEDBACK_ID NUMBER NOT NULL PRIMARY KEY,  
    STARS_NO NUMBER(1,0) NOT NULL  
);
```

-- B) CREATING OTHER TABLES:

--8) CREATING USER_TBL

```
CREATE TABLE USER_TBL(  

```

```
USER_ID    NUMBER NOT NULL PRIMARY KEY,
F_NAME     VARCHAR2(30) NOT NULL,
L_NAME     VARCHAR2(30) NOT NULL,
U_EMAIL    VARCHAR2(50) NOT NULL,
PASSWORD   VARCHAR2(20) NOT NULL,
U_PICTURE  BLOB
);

--9)CREATING DOB_TBL
CREATE TABLE DOB_TBL(
    DOB_ID    NUMBER NOT NULL PRIMARY KEY,
    DOB       DATE NOT NULL,
    PRIVACY_ID NUMBER NOT NULL,
    USER_ID  NUMBER NOT NULL
);

--10)CREATING USER_ROLE_TBL
CREATE TABLE USER_ROLE_TBL(
    U_ROLE_ID NUMBER NOT NULL PRIMARY KEY,
    ROLE_ID  NUMBER NOT NULL,
    USER_ID  NUMBER NOT NULL
);
```

--11)CREATING USER_LOCATION_TBL

CREATE TABLE USER_LOCATION_TBL(

U_LOCATION_ID NUMBER NOT NULL PRIMARY KEY,

U_COUNTRY VARCHAR2(30) NOT NULL,

U_STATE VARCHAR2(30) NOT NULL,

U_CITY VARCHAR2(50) NOT NULL,

PRIVACY_ID NUMBER NOT NULL,

USER_ID NUMBER NOT NULL

);

--12)CREATING WORK_TBL

CREATE TABLE WORK_TBL(

WORK_ID NUMBER NOT NULL PRIMARY KEY,

COMPANY_NAME VARCHAR2(100) NOT NULL,

POSITION VARCHAR2(50) NOT NULL,

WORK_COUNTRY VARCHAR2(30) NOT NULL,

WORK_STATE VARCHAR2(30) NOT NULL,

WORK_CITY VARCHAR2(50) NOT NULL,

S_DATE DATE,

E_DATE DATE,

PRIVACY_ID NUMBER NOT NULL,

USER_ID NUMBER NOT NULL

);

--13) CREATING USER_INTEREST_TBL

CREATE TABLE USER_INTEREST_TBL(

U_INTEREST_ID NUMBER NOT NULL PRIMARY KEY,

INTEREST_ID NUMBER NOT NULL,

USER_ID NUMBER NOT NULL

);

--14) CREATING PHONE_TBL

CREATE TABLE PHONE_TBL(

PHONE_ID NUMBER NOT NULL PRIMARY KEY,

PHONE NUMBER NOT NULL,

PRIVACY_ID NUMBER NOT NULL,

USER_ID NUMBER NOT NULL

);

--15) CREATING LINK_TBL

CREATE TABLE LINK_TBL(

LINK_ID NUMBER NOT NULL PRIMARY KEY,

LINK VARCHAR2(100) NOT NULL,

PRIVACY_LINK_ID NUMBER NOT

NULL,

```
USER_LINK_ID NUMBER NOT NULL
);
```

```
--16) CREATING EDUCATION_TBL
```

```
CREATE TABLE EDUCATION_TBL(
    EDUCATION_ID      NUMBER NOT NULL PRIMARY KEY,
    INSTITUTION_NAME  VARCHAR2(50) NOT NULL,
    EDUCATION_COUNTRY VARCHAR2(30) NOT NULL,
    EDUCATION_STATE   VARCHAR2(30) NOT NULL,
    EDUCATION_CITY    VARCHAR2(50) NOT NULL,
    S_DATE            DATE,
    E_DATE            DATE,
    MAJOR_ED_ID       NUMBER NOT NULL,
    DEGREE_ED_ID      NUMBER NOT NULL,
    USER_ID           NUMBER NOT NULL,
    PRIVACY_ID        NUMBER NOT NULL
);
```

```
--17) CREATING USER_SKILL_TBL
```

```
CREATE TABLE USER_SKILL_TBL(
    U_SKILL_ID        NUMBER NOT NULL PRIMARY KEY,
    SKILL_FK_ID       NUMBER NOT NULL,
```

```
USER_FK_ID NUMBER NOT NULL
```

```
);
```

```
--18) CREATING FOLLOWING_TBL
```

```
CREATE TABLE FOLLOWING_TBL(
```

```
    FOLLOWING_ID    NUMBER NOT NULL PRIMARY KEY,
```

```
                    FOLLOWER_ID NUMBER NOT NULL,
```

```
                    FOLLOWED_ID NUMBER NOT NULL
```

```
);
```

```
--19) CREATING ENDORSE_TBL
```

```
CREATE TABLE ENDORSE_TBL(
```

```
    ENDORSE_ID NUMBER NOT NULL PRIMARY KEY,
```

```
    ENDORSE_DATE DATE NOT NULL,
```

```
    APPROVEMENT_STATUS VARCHAR2(3),
```

```
    SKILL_ID NUMBER NOT NULL,
```

```
    ENDORSER_ID NUMBER NOT NULL,
```

```
    ENDORSED_ID NUMBER NOT NULL
```

```
);
```

```
--20) CREATING PROJECT_TBL
```

```
CREATE TABLE PROJECT_TBL(
```

```
    PROJECT_ID    NUMBER NOT NULL PRIMARY KEY,
```

```
PROJECT_TITLE    VARCHAR2(50),  
  
EXPLANATION  
  
VARCHAR2(250),  
  
UPDATE_TIME     DATE,  
PRIVACY_ID      NUMBER NOT NULL,  
PROJECT_OWNER   NUMBER NOT  
  
NULL  
);
```

```
--21) CREATING USER_PROJECT_TBL
```

```
CREATE TABLE USER_PROJECT_TBL(  
    USER_PROJECT_ID NUMBER NOT NULL PRIMARY KEY,  
  
    ROLE_ID NUMBER NOT NULL,  
    USER_ID NUMBER NOT NULL,  
    PROJECT_ID NUMBER NOT NULL  
);
```

```
--22) CREATING PROJECT_SKILL_TBL
```

```
CREATE TABLE PROJECT_SKILL_TBL(  
    PROJECT_SKILL_ID NUMBER NOT NULL PRIMARY KEY,  
  
    SKILL_ID NUMBER NOT NULL,  
    PROJECT_ID NUMBER NOT NULL  
);
```

--23) CREATING PROJECT_INTEREST_TBL

CREATE TABLE PROJECT_INTEREST_TBL(

PROJECT_INTEREST_ID NUMBER NOT NULL PRIMARY KEY,

INTEREST_ID NUMBER NOT NULL,

PROJECT_ID NUMBER NOT NULL

);

--24) CREATING STEP_TBL

CREATE TABLE STEP_TBL(

STEP_ID NUMBER NOT NULL PRIMARY KEY,

STEP_NAME VARCHAR2(20) NOT NULL,

SHARE_PERCENT INT NOT NULL,

COMPLETION_PERCENT INT NOT NULL,

UPDATE_DATE DATE,

S_DATE DATE,

E_DATE DATE,

PROJECT_ID NUMBER NOT NULL

);

--25) CREATING STEP_USER_TBL

CREATE TABLE STEP_USER_TBL(

STEP_USER_ID NUMBER NOT NULL PRIMARY KEY,

```
USER_ID NUMBER NOT NULL,  
STEP_ID NUMBER NOT NULL  
  
);  
  
--26) CREATING FEEDBACK_USER_PROJECT_TBL  
CREATE TABLE FEEDBACK_USER_PROJECT_TBL(  
    FUP_ID NUMBER NOT NULL PRIMARY KEY,  
    PRIVACY_ID NUMBER NOT NULL,  
    FEEDBACK_ID NUMBER NOT NULL,  
    GIVER_ID NUMBER NOT NULL,  
    RECEIVER_ID NUMBER NOT NULL,  
    PROJECT_ID NUMBER NOT NULL  
  
);  
  
--27) CREATING IDEA_TBL  
CREATE TABLE IDEA_TBL(  
    IDEA_ID          NUMBER NOT NULL PRIMARY KEY,  
    IDEA_TITLE       VARCHAR2(20) NOT NULL,  
    IDEA_EXPL        VARCHAR2(300) NOT NULL,  
    IDEA_DATE        DATE,  
    PUBLISH_STATUS   VARCHAR2(3) NOT NULL,  
    INUSE_STATUS     VARCHAR2(3)  
  
NOT NULL,
```

```
PRIVACY_ID NUMBER NOT NULL,  
USER_ID NUMBER NOT NULL  
);  
  
--28)CREATING IDEA_PROJECT_TBL  
CREATE TABLE IDEA_PROJECT_TBL(  
    IDEA_PROJECT_ID NUMBER NOT NULL PRIMARY KEY,  
    PROJECT_ID NUMBER NOT NULL,  
    IDEA_ID NUMBER NOT NULL  
);  
  
--29)CREATING IDEA_SKILL_TBL  
CREATE TABLE IDEA_SKILL_TBL(  
    IDEA_SKILL_ID NUMBER NOT NULL PRIMARY KEY,  
    SKILL_ID NUMBER NOT NULL,  
    IDEA_ID NUMBER NOT NULL  
);  
  
--30)CREATING IDEA_INTEREST_TBL  
CREATE TABLE IDEA_INTEREST_TBL(  
    IDEA_INTEREST_ID NUMBER NOT NULL PRIMARY KEY,  
    INTEREST_ID NUMBER NOT NULL,  
    IDEA_ID NUMBER NOT NULL
```

);

--31) CREATING COMMENT_TBL

CREATE TABLE COMMENT_TBL(

COMMENT_ID NUMBER NOT NULL PRIMARY KEY,

COMMENT_CONTENT VARCHAR2(300) NOT NULL,

COMMENT_DATE DATE,

USER_ID NUMBER NOT NULL,

COMMENT_FK NUMBER,

IDEA_FK NUMBER,

PROJECT_FK NUMBER

);

-----CREATING RELATIONSHIPS/FOREIGN KEYS-----

--DOB_TBL FKS

ALTER TABLE DOB_TBL

ADD CONSTRAINT PRIVACY_ID FOREIGN KEY (PRIVACY_ID)

REFERENCES PRIVACY_TBL (PRIVACY_ID);

ALTER TABLE DOB_TBL

ADD CONSTRAINT USER_ID FOREIGN KEY (USER_ID)

REFERENCES USER_TBL (USER_ID);


```
--USER_ROLE_TBL FKS
```

```
ALTER TABLE USER_ROLE_TBL
```

```
ADD CONSTRAINT ROLE_ID FOREIGN KEY (ROLE_ID)
```

```
REFERENCES ROLE_TBL (ROLE_ID);
```

```
ALTER TABLE USER_ROLE_TBL
```

```
ADD CONSTRAINT USER_FK_UR FOREIGN KEY (USER_ID)
```

```
REFERENCES USER_TBL (USER_ID);
```

```
--USER_LOCATION_TBL FKS
```

```
ALTER TABLE USER_LOCATION_TBL
```

```
ADD CONSTRAINT PRIVACY_FK_UL FOREIGN KEY (PRIVACY_ID)
```

```
REFERENCES PRIVACY_TBL (PRIVACY_ID);
```

```
ALTER TABLE USER_LOCATION_TBL
```

```
ADD CONSTRAINT USER_FK_UL FOREIGN KEY (USER_ID)
```

```
REFERENCES USER_TBL (USER_ID);
```

```
--WORK_TBL FKS
```

```
ALTER TABLE WORK_TBL
```

```
ADD CONSTRAINT PRIVACY_FK_W FOREIGN KEY (PRIVACY_ID)
REFERENCES PRIVACY_TBL (PRIVACY_ID);
```

```
ALTER TABLE WORK_TBL
```

```
ADD CONSTRAINT USER_FK_W FOREIGN KEY (USER_ID)
REFERENCES USER_TBL (USER_ID);
```

```
--USER_INTEREST_TBL FKS
```

```
ALTER TABLE USER_INTEREST_TBL
ADD CONSTRAINT INTEREST_ID FOREIGN KEY (INTEREST_ID)
REFERENCES INTEREST_AREA_TBL (INTEREST_ID);
```

```
ALTER TABLE USER_INTEREST_TBL
ADD CONSTRAINT USER_FK_UIN FOREIGN KEY (USER_ID)
REFERENCES USER_TBL (USER_ID);
```

```
--PHONE_TBL FKS
```

```
ALTER TABLE PHONE_TBL
ADD CONSTRAINT PRIVACY_FK_PH FOREIGN KEY (PRIVACY_ID)
REFERENCES PRIVACY_TBL (PRIVACY_ID);
```

```
ALTER TABLE PHONE_TBL
```

```
ADD CONSTRAINT USER_FK_PH FOREIGN KEY (USER_ID)
```

```
REFERENCES USER_TBL (USER_ID);
```

```
--LINK_TBL FKS
```

```
ALTER TABLE LINK_TBL
```

```
ADD CONSTRAINT LINK_P_FK FOREIGN KEY (PRIVACY_LINK_ID)
```

```
REFERENCES PRIVACY_TBL (PRIVACY_ID);
```

```
ALTER TABLE LINK_TBL
```

```
ADD CONSTRAINT LINK_U_FK FOREIGN KEY (USER_LINK_ID)
```

```
REFERENCES USER_TBL (USER_ID);
```

```
--EDUCATION_TBL FKS
```

```
ALTER TABLE EDUCATION_TBL
```

```
ADD CONSTRAINT MAJOR_FK_ED FOREIGN KEY (MAJOR_ED_ID)
```

```
REFERENCES MAJOR_TBL (MAJOR_ID);
```

```
ALTER TABLE EDUCATION_TBL
```

```
ADD CONSTRAINT DEGREE_FK_ED FOREIGN KEY (DEGREE_ED_ID)
```

```
REFERENCES DEGREE_TBL (DEGREE_ID);
```

```
ALTER TABLE EDUCATION_TBL  
ADD CONSTRAINT FK_PRIVACY FOREIGN KEY (PRIVACY_ID)  
REFERENCES PRIVACY_TBL (PRIVACY_ID);
```

```
ALTER TABLE EDUCATION_TBL  
ADD CONSTRAINT FK_USER FOREIGN KEY (USER_ID)  
REFERENCES USER_TBL (USER_ID);
```

```
--USER_SKILL_TBL FKS  
ALTER TABLE USER_SKILL_TBL  
ADD CONSTRAINT FK_USERSKILL FOREIGN KEY (SKILL_FK_ID)  
REFERENCES SKILL_TBL (SKILL_ID);
```

```
ALTER TABLE USER_SKILL_TBL  
ADD CONSTRAINT FK_SKILL_USER FOREIGN KEY (USER_FK_ID)  
REFERENCES USER_TBL (USER_ID);
```

```
--FOLLOWING_TBL FKS  
ALTER TABLE FOLLOWING_TBL  
ADD CONSTRAINT FK_FOLLOWER FOREIGN KEY (FOLLOWER_ID)  
REFERENCES USER_TBL (USER_ID);
```

```
ALTER TABLE FOLLOWING_TBL
ADD CONSTRAINT FK_FOLLOWED FOREIGN KEY (FOLLOWED_ID)
REFERENCES USER_TBL (USER_ID);
```

```
--ENDORSE_TBL FKS
```

```
ALTER TABLE ENDORSE_TBL
ADD CONSTRAINT FK_SKILL FOREIGN KEY (SKILL_ID)
REFERENCES SKILL_TBL (SKILL_ID);
```

```
ALTER TABLE ENDORSE_TBL
ADD CONSTRAINT FK_ENDORSER FOREIGN KEY (ENDORSER_ID)
REFERENCES USER_TBL (USER_ID);
```

```
ALTER TABLE ENDORSE_TBL
ADD CONSTRAINT FK_ENDORSED FOREIGN KEY (ENDORSED_ID)
REFERENCES USER_TBL (USER_ID);
```

```
--USER_PROJECT_TBL FKS
```

```
ALTER TABLE USER_PROJECT_TBL
ADD CONSTRAINT FK_ROLE FOREIGN KEY (ROLE_ID)
REFERENCES ROLE_TBL (ROLE_ID);
```

```
ALTER TABLE USER_PROJECT_TBL
ADD CONSTRAINT FK_U_PROJECT FOREIGN KEY (PROJECT_ID)
REFERENCES PROJECT_TBL (PROJECT_ID);
```

```
ALTER TABLE USER_PROJECT_TBL
ADD CONSTRAINT FK_PROJECT_MEMBER FOREIGN KEY (USER_ID)
REFERENCES USER_TBL (USER_ID);
```

```
ALTER TABLE USER_PROJECT_TBL
ADD CONSTRAINT FK_PRIVACY_OF_PROJECT FOREIGN KEY (PRIVACY_ID)
REFERENCES PRIVACY_TBL (PRIVACY_ID);
```

```
--PROJECT_SKILL_TBL FKS
```

```
ALTER TABLE PROJECT_SKILL_TBL
ADD CONSTRAINT FK_PROJECT_SKILLS FOREIGN KEY (SKILL_ID)
REFERENCES SKILL_TBL (SKILL_ID);
```

```
ALTER TABLE PROJECT_SKILL_TBL
ADD CONSTRAINT FK_PROJECT FOREIGN KEY (PROJECT_ID)
REFERENCES PROJECT_TBL (PROJECT_ID);
```

```
--PROJECT_INTEREST_TBL FKS
```

```
ALTER TABLE PROJECT_INTEREST_TBL
```

```
ADD CONSTRAINT FK_INTEREST_IN_PI FOREIGN KEY (INTEREST_ID)
```

```
REFERENCES INTEREST_AREA_TBL (INTEREST_ID);
```

```
ALTER TABLE PROJECT_INTEREST_TBL
```

```
ADD CONSTRAINT FK_PROJECT_IN_PI FOREIGN KEY (PROJECT_ID)
```

```
REFERENCES PROJECT_TBL (PROJECT_ID);
```

```
--STEP_TBL FKS
```

```
ALTER TABLE STEP_TBL
```

```
ADD CONSTRAINT FK_PROJECT_STEPS FOREIGN KEY (PROJECT_ID)
```

```
REFERENCES PROJECT_TBL (PROJECT_ID);
```

```
--STEP_USER_TBL FKS
```

```
ALTER TABLE STEP_USER_TBL
```

```
ADD CONSTRAINT FK_USER_IN_SU FOREIGN KEY (USER_ID)
```

```
REFERENCES USER_TBL (USER_ID);
```

```
ALTER TABLE STEP_USER_TBL
```

```
ADD CONSTRAINT FK_STEP_IN_SU FOREIGN KEY (STEP_ID)
```

```
REFERENCES STEP_TBL (STEP_ID);
```

```
--FEEDBACK_USER_PROJECT_TBL FKS
```

```
ALTER TABLE FEEDBACK_USER_PROJECT_TBL
```

```
ADD CONSTRAINT FK_PRIVACY_IN_FUP FOREIGN KEY (PRIVACY_ID)
```

```
REFERENCES PRIVACY_TBL (PRIVACY_ID);
```

```
ALTER TABLE FEEDBACK_USER_PROJECT_TBL
```

```
ADD CONSTRAINT FK_FEEDBACK_IN_FUP FOREIGN KEY (FEEDBACK_ID)
```

```
REFERENCES FEEDBACK_TBL (FEEDBACK_ID);
```

```
ALTER TABLE FEEDBACK_USER_PROJECT_TBL
```

```
ADD CONSTRAINT FK_GIVER FOREIGN KEY (GIVER_ID)
```

```
REFERENCES USER_TBL (USER_ID);
```

```
ALTER TABLE FEEDBACK_USER_PROJECT_TBL
```

```
ADD CONSTRAINT FK_RECEIVER FOREIGN KEY (RECEIVER_ID)
```

```
REFERENCES USER_TBL (USER_ID);
```

```
ALTER TABLE FEEDBACK_USER_PROJECT_TBL
```

```
ADD CONSTRAINT FK_FUP FOREIGN KEY (PROJECT_ID)
```

```
REFERENCES PROJECT_TBL (PROJECT_ID);
```



```
--IDEA_TBL FKS
```

```
ALTER TABLE IDEA_TBL
```

```
ADD CONSTRAINT FK_PRIVACY_IN_IDEA FOREIGN KEY (PRIVACY_ID)
```

```
REFERENCES PRIVACY_TBL (PRIVACY_ID);
```

```
ALTER TABLE IDEA_TBL
```

```
ADD CONSTRAINT FK_USER_IN_IDEA FOREIGN KEY (USER_ID)
```

```
REFERENCES USER_TBL (USER_ID);
```

```
--DELETE THIS ONE => ALTER TABLE table_name
```

```
--DROP CONSTRAINT constraint_name;
```

```
--ALTER TABLE IDEA_TBL
```

```
--ADD CONSTRAINT FK_PROJECT_BASED_ON_THIS FOREIGN KEY (PROJECT_ID)
```

```
--REFERENCES PROJECT_TBL (PROJECT_ID);
```

```
--IDEA_PROJECT_TBL FKS
```

```
ALTER TABLE IDEA_PROJECT_TBL
```

```
ADD CONSTRAINT FK_PROJECT_IN_IF FOREIGN KEY (PROJECT_ID)
```

```
REFERENCES PROJECT_TBL (PROJECT_ID);
```

```
ALTER TABLE IDEA_PROJECT_TBL
```

```
ADD CONSTRAINT FK_IDEA_IN_IF FOREIGN KEY (IDEA_ID)
```

```
REFERENCES IDEA_TBL (IDEA_ID);
```

```
--IDEA_SKILL_TBL FKS
```

```
ALTER TABLE IDEA_SKILL_TBL
```

```
ADD CONSTRAINT FK_SKILL_IN_IS FOREIGN KEY (SKILL_ID)
```

```
REFERENCES SKILL_TBL (SKILL_ID);
```

```
ALTER TABLE IDEA_SKILL_TBL
```

```
ADD CONSTRAINT FK_IDEA_IN_IS FOREIGN KEY (IDEA_ID)
```

```
REFERENCES IDEA_TBL (IDEA_ID);
```

```
--IDEA_INTEREST_TBL FKS
```

```
ALTER TABLE IDEA_INTEREST_TBL
```

```
ADD CONSTRAINT FK_IDEA_IN_II FOREIGN KEY (IDEA_ID)
```

```
REFERENCES IDEA_TBL (IDEA_ID);
```

```
ALTER TABLE IDEA_INTEREST_TBL
```

```
ADD CONSTRAINT FK_INTEREST_IN_II FOREIGN KEY (INTEREST_ID)
```

```
REFERENCES INTEREST_AREA_TBL (INTEREST_ID);
```

--COMMENT_TBL FKS

ALTER TABLE COMMENT_TBL

ADD CONSTRAINT FK_COMMENT_WRITER FOREIGN KEY (USER_ID)

REFERENCES USER_TBL (USER_ID);

ALTER TABLE COMMENT_TBL

ADD CONSTRAINT FK_COMMENT_OF_COM FOREIGN KEY (COMMENT_FK)

REFERENCES COMMENT_TBL (COMMENT_ID);

ALTER TABLE COMMENT_TBL

ADD CONSTRAINT FK_IDEA_OF_COM FOREIGN KEY (IDEA_FK)

REFERENCES IDEA_TBL (IDEA_ID);

ALTER TABLE COMMENT_TBL

ADD CONSTRAINT FK_PROJECT_OF_COM FOREIGN KEY (PROJECT_FK)

REFERENCES PROJECT_TBL (PROJECT_ID);

--TO SHOW ALL TABLES OWNED BY THIS USER:

select tablespace_name, table_name from user_tables;

COMMIT;

E-Table Populating

--FILLING OUT TABLES

```
INSERT INTO PRIVACY_TBL (PRIVACY_ID, PRIVACY_NAME)
```

```
VALUES (1, 'private');
```

```
INSERT INTO PRIVACY_TBL (PRIVACY_ID, PRIVACY_NAME)
```

```
VALUES (2, 'public');
```

```
INSERT INTO PRIVACY_TBL (PRIVACY_ID, PRIVACY_NAME)
```

```
VALUES (3, 'followers');
```

```
INSERT INTO PRIVACY_TBL (PRIVACY_ID, PRIVACY_NAME)
```

```
VALUES (4, 'custom');
```

```
INSERT INTO PRIVACY_TBL (PRIVACY_ID, PRIVACY_NAME)
```

```
VALUES (5, 'team');
```

```
INSERT INTO INTEREST_AREA_TBL (INTEREST_ID, INTEREST_NAME, INTEREST_EXPL)
```

```
VALUES (1, 'Conventional', 'Conventional people like organization,  
structure, and stability. They prefer work that requires attention  
to detail and routine.');
```

```
INSERT INTO INTEREST_AREA_TBL (INTEREST_ID, INTEREST_NAME, INTEREST_EXPL)
```

```
VALUES (2, 'Investigative', 'Investigative types enjoy solving  
complex problems and appreciate abstract ideas. Science, math,  
and engineering are areas in which investigative types usually excel.');
```

```
INSERT INTO INTEREST_AREA_TBL (INTEREST_ID, INTEREST_NAME, INTEREST_EXPL)
```

```
VALUES (3, 'Social', 'Social types value relationships and gravitate
```

towards the “helping” professions such as nursing, counseling and teaching. They enjoy working with people and communicating.');

```
INSERT INTO INTEREST_AREA_TBL (INTEREST_ID, INTEREST_NAME, INTEREST_EXPL)
VALUES (4, 'Artistic', 'Creativity expression is highly valued by artistic
types. They prefer work that does not follow a clear set of rules or
regulations.');
```

```
INSERT INTO INTEREST_AREA_TBL (INTEREST_ID, INTEREST_NAME, INTEREST_EXPL)
VALUES (5, 'Realistic', 'Realistic types enjoy working outdoors and prefer
work that is hands-on. They like using machinery or tools and would
probably not enjoy most office jobs.');
```

```
INSERT INTO INTEREST_AREA_TBL (INTEREST_ID, INTEREST_NAME, INTEREST_EXPL)
VALUES (6, 'Enterprising', 'Enterprising types like to start and carry out
projects. They do not mind taking risks and are well-suited for leadership
roles.');
```

```
INSERT INTO USER_TBL (USER_ID, F_NAME, L_NAME, U_EMAIL, PASSWORD, U_PICTURE,
ACTIVE)
```

```
VALUES (1, 'Fatemeh', 'Khastkhoda', 'fkhastkhodaard@eiu.edu', 'fA1#', '', 1);
```

```
INSERT INTO USER_TBL (USER_ID, F_NAME, L_NAME, U_EMAIL, PASSWORD, U_PICTURE,
ACTIVE)
```

```
VALUES (2, 'Mark', 'Zuckerberg', 'mzuckerberg@fb.com', 'fAce@2', '', 1);
```

```
INSERT INTO USER_TBL (USER_ID, F_NAME, L_NAME, U_EMAIL, PASSWORD, U_PICTURE,
ACTIVE)
```

```
VALUES (3, 'Larry', 'Page', 'lpage@gmail.com', 'LaRp#', '', 1);
```

```
INSERT INTO ROLE_TBL (ROLE_ID, ROLE_NAME)
```

```
VALUES (1, 'member');
```

```
INSERT INTO ROLE_TBL (ROLE_ID, ROLE_NAME)
```

```
VALUES (2, 'leader');
```

```
INSERT INTO ROLE_TBL (ROLE_ID, ROLE_NAME)
```

```
VALUES (3, 'owner');
```

```
INSERT INTO ROLE_TBL (ROLE_ID, ROLE_NAME)
```

```
VALUES (4, 'admin');
```

```
INSERT INTO USER_ROLE_TBL (U_ROLE_ID, ROLE_ID, USER_ID)
```

```
VALUES (1, 1, 1);
```

```
INSERT INTO USER_ROLE_TBL (U_ROLE_ID, ROLE_ID, USER_ID)
```

```
VALUES (2, 4, 1);
```

```
INSERT INTO USER_ROLE_TBL (U_ROLE_ID, ROLE_ID, USER_ID)
```

```
VALUES (3, 2, 2);
```

```
INSERT INTO USER_ROLE_TBL (U_ROLE_ID, ROLE_ID, USER_ID)
```

```
VALUES (4, 1, 3);
```

```
INSERT INTO USER_ROLE_TBL (U_ROLE_ID, ROLE_ID, USER_ID)
```

```
VALUES (5, 3, 2);
```

```
INSERT INTO USER_ROLE_TBL (U_ROLE_ID, ROLE_ID, USER_ID)
```

```
VALUES (6, 1, 2);
```

```
INSERT INTO PROJECT_TBL (PROJECT_ID, PROJECT_TITLE, EXPLANATION,
UPDATE_TIME)
VALUES (1, 'Project Website', 'The project is about a website...', TO_DATE('2015/07/01
21:02:44', 'yyyy/mm/dd hh24:mi:ss'), 5);
INSERT INTO PROJECT_TBL (PROJECT_ID, PROJECT_TITLE, EXPLANATION, UPDATE_TIME,
PRIVACY_ID)
VALUES (2, 'Emoji Effects on Users', 'This research is about the effects of emoji on users in
different emotions', TO_DATE('2015/10/08 21:02:44', 'yyyy/mm/dd hh24:mi:ss'), 2);

INSERT INTO USER_PROJECT_TBL (USER_PROJECT_ID, USER_ID, PROJECT_ID, ROLE_ID,
PRIVACY_ID)
VALUES (1, 1, 1, 4, 5);
INSERT INTO USER_PROJECT_TBL (USER_PROJECT_ID, USER_ID, PROJECT_ID, ROLE_ID,
PRIVACY_ID)
VALUES (2, 2, 2, 3, 2);
INSERT INTO USER_PROJECT_TBL (USER_PROJECT_ID, USER_ID, PROJECT_ID, ROLE_ID,
PRIVACY_ID)
VALUES (3, 1, 2, 2, 2);

INSERT INTO FOLLOWING_TBL (FOLLOWING_ID, FOLLOWER_ID, FOLLOWED_ID)
VALUES (1, 1, 2);
INSERT INTO FOLLOWING_TBL (FOLLOWING_ID, FOLLOWER_ID, FOLLOWED_ID)
```

```
VALUES (2, 1, 3);
```

```
INSERT INTO FOLLOWING_TBL (FOLLOWING_ID, FOLLOWER_ID, FOLLOWED_ID)
```

```
VALUES (3, 2, 1);
```

```
INSERT INTO FOLLOWING_TBL (FOLLOWING_ID, FOLLOWER_ID, FOLLOWED_ID)
```

```
VALUES (4, 3, 2);
```