UNIVERSITY OF
CALGARY

Project Report

Research My Professor

CPSC 471 Spring 2018

Group 2:
Jonathan Yee
Brandt Davis
Dylan Loader

June 30th, 2018

Contents

1 Abstract

Research is one of the most important functions of post secondary institutions. Students themselves can become involved in research but, there are some hurdles to overcome. ResearchMyProf aims to alleviate the burdens placed on students and researchers in the early stages of research. Before they can start any research students look for a supervising professor, or researchers look for collaborators to help with their research. Currently information about professors and researchers is distributed on the internet, but there is no central location. Most post secondary institutions have a database of all their professors and researchers but they vary in the accuracy and amount of detail. ResearchMyProf aims to combat this issue. The solution is centered around a database management system that interacts with users through a webpage. This system will allow users to add information on professors and researchers around the world. Users can then search the database through the website to find professors and/or researchers that they could work with on research. This method is much less time consuming than navigating institution websites which vary in quality. ResearchMyProf will make starting research easier thus benefiting students, professors, researchers and post secondary institutions as a whole.

2 Introduction

One of the important functions of post secondary institutions is research. In 2016 Universities consisting of over 100 full-time teaching staff employed over 45,000 staff involved in research [1]. Students of post secondary institutions are encouraged to participate in academic research by working with a professor who would be their supervisor[2]. Supervisors provide support and mentoring to students so that they can complete their research and publish their findings.

One of the key steps in academic research is the student's choice of supervisor. To students that are just beginning to engage or who are furthering their research, this can be an overwhelming issue. Without a supervisor to help guide them, students hoping to pursue research are often unable to apply for grants, and may become dissuaded from applying because many application sites are outdated poorly maintained [3]. Finding a supervisor with relevant abilities, background, and research interest increases the chances that academic collaboration will benefit the student and supervisor. Often, researching professors publish in areas that differ from their academic credentials, or the scope of their current research differs significantly from their formal training. This discrepancy can often result in a mismatch of the abilities and interests of the students who approach them. Information about professors can be found on their personal pages and their publications can also be searched for but, the information can be out of date or incorrect [4].

What our project will do is create a crowdsourced database which would allow users to record professor's competencies, background,and research interests and publications. At the same time the database would be moderated by specified users to minimize inaccuracies and avoid slanderous accusations [5].

2.1 Problem Definition

Students who want to become involved in or continue any research pursuits need to find a professor who would be interested in becoming their supervisor. This issue was of great interest to us, as we are students ourselves. The problem is that it is no straightforward way to find a supervisor who matches well with a student. There are databases that contain information about professors, and typically, the institution that they work for will also have a database, but these are occasionally outdated. Overall, current collaboration databases are rather general, and this leaves the student alone in the task of finding a collaborator, which can be quite daunting to someone who is new to academic research [6]. Many professors do have their research information stored on an institution's database, but, it is up to the institution to ensure that the information is correct and up to date, unfortunately this is not as reliable a solution as one may think.

2.2 The Solution

Our system is intended to be a crowdsourced alternative to the in-house implementations that are typically used to search universities for potential research supervisors. Our goal was to create a database to store current information about researchers and their work, while minimizing the subjective bias often found in crowd-sourced academic data, typically found on sites such as RateMyProfessor [7]. Our project accomplishes this bias reduction by allowing moderators to remove flagged entities, while allowing anyone the ability to add, flag, and search for data in the database.The purpose was to help students find potential supervisors while simultaneously allowing them to contribute to the database, which would have the potential

to help others. Moderators will have the same permissions as the general users but, they will also have the additional permission of reviewing flagged entities. Moderators will have the power to remove entries that are incorrect and/or not useful.

3 Project Design

3.1 Types of Users

There are three types of users in our system:

1. **General User:**
   The general users of our system will typically be researchers of some sort, whether they are amateur researchers, supervisors, or curious students, our system will help them find the information they need. In terms of how general users will use our system, there are several features that can be utilized. A general user may not only search for different profiles by country, institution, and , but they are also able to add profiles to the database. In the case that information does indeed make its way into the database, the users will be able to submit a report specifying the erroneous information, which will then be handled by the moderators.

2. **Moderators:**
   In our system the moderators will be the middle ground between general users and administrators. They will be general users with more permissions, such as viewing reports, and assessing them. Once a report has been handles, the moderator will be able mark the report as resolved, which would remove the report from the list of "active" reports, but it would not be removed from the database for record-keeping purposes. While this feature is currently not available, a future implementation of our system would include the ability to edit the profiles that users add, to streamline the report resolution process.

3. **Administrators:**
   Administrators of our system will be the users who keep the system running properly, and ensuring that users who consistently enter erroneous information will be warned, and if needed removed from the system. The administrators will be able to view all users that exist in the database, as well as their classifications(general users, moderators, and administrators) they will also be able to view the permissions of each user, and if necessary, alter the permissions as well.

3.2 Transaction Collection

Our system uses many transactions across each page to achieve the functionality that we have:

- **Login Page**
  The login page will take user input consisting of a username and password, and will check of course, if the input valid, as well as(assuming the inputs are valid) a user ID, and administrator and moderator flags, to determine which pages the user will see. If the user input is not valid(the username or password do not match any entry of our database, then the page will display an error message indicating that one or more of the input items is invalid.

- **Administration Page**

  1. **Retrieve User Information**
     While there is no direct user input in this transaction, the system still retrieves the user id, the username, the user mod and admin flags, and the start date of the user. This is so administrators can view the users on the site.
  2. **Reverse Moderator Status**
     With this transaction, input is automated, based on the selection of the user. The system will take a specified user's "user id", and uses it to either revoke or enable moderator permissions, based on the initial state of the user's permissions. If an administrator clicks on the corresponding button, they can revoke or enable a users privileges as a moderator.
  3. **Reverse Administrator Status**
     With this transaction, input is automated as well, and again, it is based on the selection of the

user. The system takes a specified user's "user id", and uses it to revoke or enable administrator permissions based on the initial state of the user's permissions. If an administrator clicks on the corresponding button, they can revoke or enable a user's privileges as an administrator.

- **Moderation Page**

  1. **Retrieve All Reports**
     There is no direct input required by this transaction, the page simply retrieves all the active reports in the database, and their corresponding attributes, which are visible in our RM diagram in the implementation section.
  2. **Mark a Report as Resolved**
     This transaction uses the data stored in the report(profile id, user id, and the submission date) as input, and simply inputs the date and time that the resolution took place.

- **Profile Page**

  1. **Retrieve Profile Name**
     The user input is not direct in this transaction, the profile id is selected when the user clicks on a profile to view. The system uses the profile id to find a corresponding profile name, and then displays it for the user to see.
  2. **Retrieve Location Information of a Profile**
     The user input is not direct in this transaction, the profile id is selected when the user clicks on a profile to view. The system uses the profile id to find a corresponding institution, postal code, country and city.
  3. **Retrieve the Topics of Interest of a profile**
     The user input is not direct in this transaction, the profile id is selected when the user clicks on a profile to view. The system uses the profile id to find corresponding topics.
  4. **Retrieve Publications that the Profile has Authored**
     The user input is not direct in this transaction, the profile id is selected when the user clicks on a profile to view. The system uses the profile id to find corresponding publications.
  5. **Retrieve Associated Profiles that the Profile has Worked With**
     The user input is not direct in this transaction, the profile IDA is selected when the user selects a profile, and the system displays any profiles that the profile in question are associated with.

- **Create Profile Page**

  1. **Insert a New Profile**
     In this transaction, the user inputs a name, and institution, and the system takes in their user id as well. This results in a new "profile" entry in the database.
  2. **Retrieve Profile ID**
     In this transaction, the user inputs a name, and institution, and the system takes in their user id as well. The output is a profile id.
  3. **Insert New Values into the "Interested in" Table**
     The system takes in a topic name from the user and a profile id(indirectly), and adds a new entry to the "Interested in" table, which consists of a profile(a researcher) and a topic that they are associated with.
  4. **Insert New Values into the "worked With" Table**
     The system takes in the id value of the newly created profile along with the profile that is specified by the user. The "worked with" table is then updated and stores the two values as an entry.
  5. **Insert New Values into the "Publication" Table**
     The system takes in a publication name and a link to the publication, and creates a new entry consisting of the name and link in the "publication" table
  6. **Insert New Values into the "Related to" table**
     The system takes in a topic which is specified by the user choosing from a list of topics and a publication name which is entered by the user. The output is an entry into the "related to" table consisting of the topic and the publication name.

7. **Insert New Values into the "Authored" table**
   The system takes in a profile id(indirectly) and a publication name which is entered by the user, and creates a new entry in the "authored" table, which consists of the profile id and publication name.
8. **Retrieve Institution Names and Postal Codes**
   The system takes in no input, but it outputs a list of institution names and their postal codes for the user to select from when creating a profile.
9. **Retrieve Topic Names**
   The system takes in no input, but it outputs a list of topic names for the user to select from when creating a profile.
10. **Retrieve Profile Names**
    The system takes in no input, but it outputs a list of profile names for the user to select from when creating a profile.

- **Report Profile Page**

1. **Add a New Report to the Database**
   The system takes in both user and profile ids(indirectly) as well as a timestamp of the current time(taken automatically) and the information about the report that was entered by the user. The system stores this information as an entry in the "report" table to be reviewed and resolved by moderators.

- **Search Page**

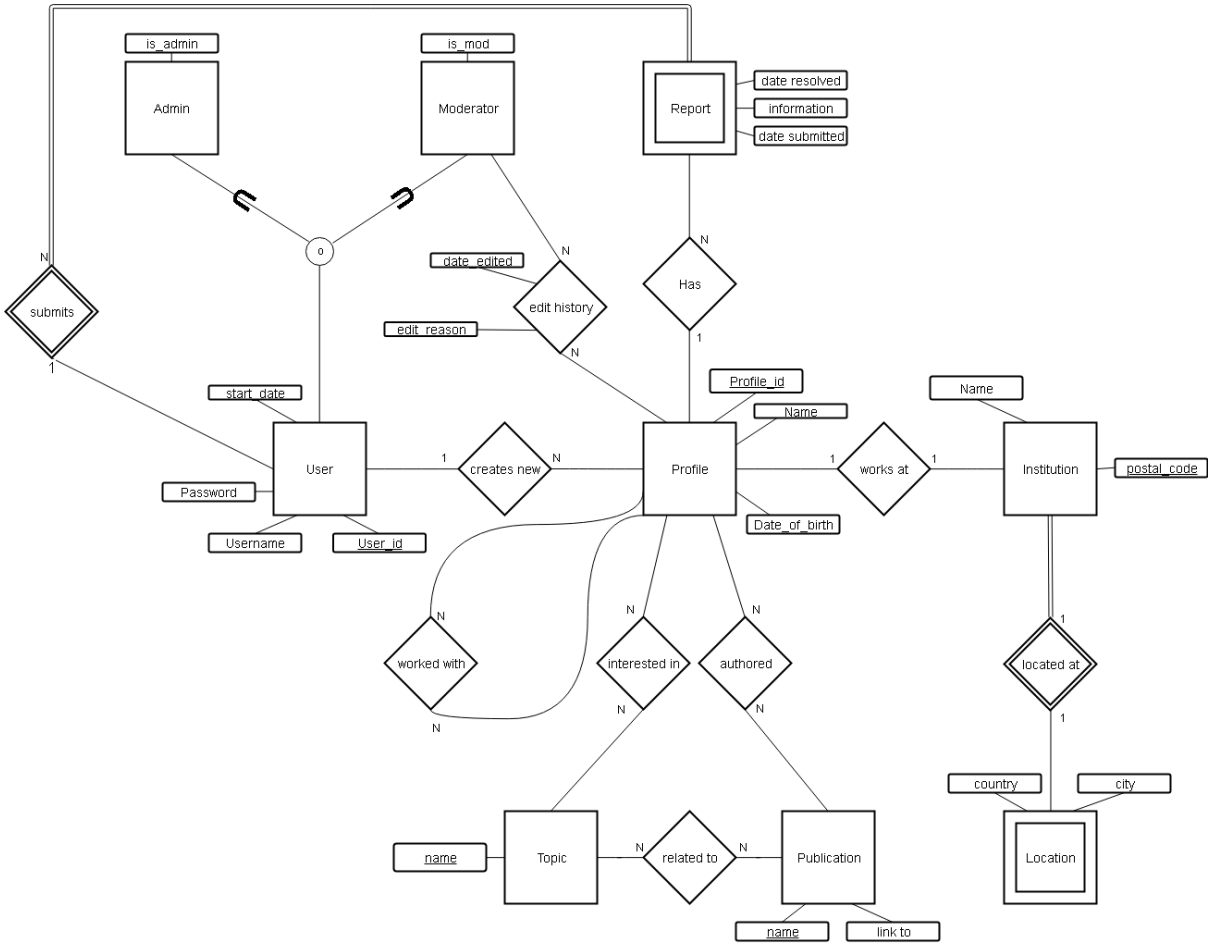
1. **Retrieve Profile id, Name, Institution, and Country**
   The system takes in a country name, which is entered by the user, and then the system outputs a list of profile names, ids, institutions, and the country that the profile is located in. This transaction is how we allow the user to search for profiles by country.
2. **Retrieve Profile id, Name, Institution, and Country**
   The system takes in an institution name, which is entered by the user, hen the system outputs a list of profile names, ids, institutions, and the country that the profile is located in. This transaction is how we allow the user to search for profiles by institution name.
3. **Retrieve a List of Profile id's that are associated with a Topic**
   The user enters a topic, and the system returns a list of profile ids that are associated with the topic.
4. **Retrieve Topics Associated with a Profile id**
   The system indirectly takes in a profile id, and returns a list of topics that the profile is associated with.
5. **Retrieve Profile id, Name, Institution, and Country**
   The system takes in a profile id indirectly, and returns the name, id, institution and country of the corresponding profile.

On every page there is a transaction with "Session", which ultimately ensures that the user is signed in by checking that the username is valid, if the user is signed in, they will not notice, however, if they sign out, they will lose access to the pages that can be viewed only if they are signed in.

3.3 Functionality

Our system utilizes several features that are fundamental to a crowdsourced website, such as allowing users to input information, and having moderators to filter/update any incorrect information. Moderators oversee any issues that general users may encounter, and if necessary they can inform administrators of repeat incidents to reduce the amount of incidents. Administrators have the most authority on the website, in order to remove any users who abuse moderator privileges or any general users who consistently enter incorrect or slanderous information.

3.4 ER Diagram

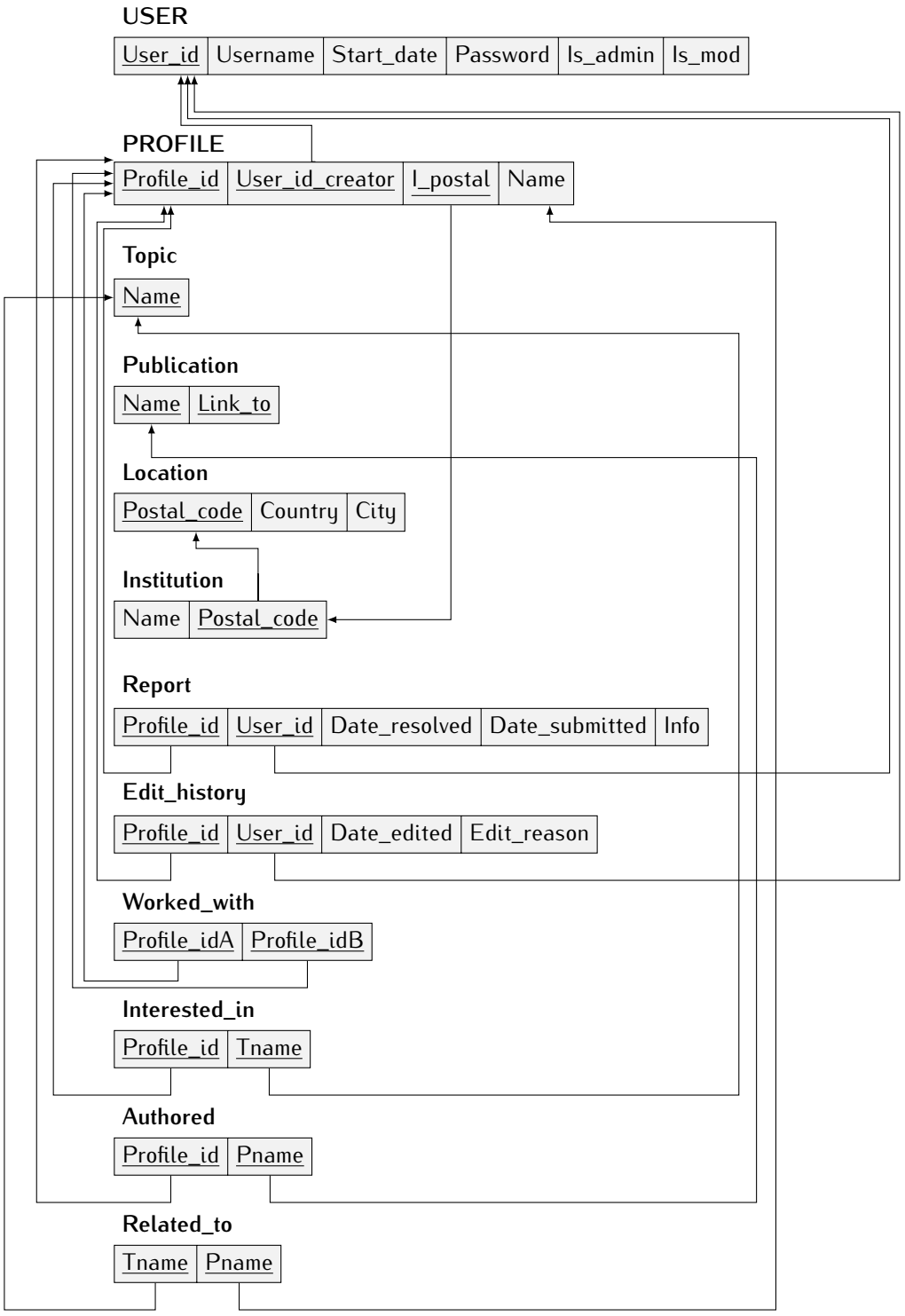


3.5 EER Diagram Changes Since the Presentation

Our EER diagram has undergone several changes since the presentation, mostly to increase the interaction among entities of our system. We implemented different user types through specialization, in order to have administrators and moderators in our system. We allowed the two new entity types of Admin and Moderator to be overlapping, so some users can be both. We also added relations from 'USER', in particular, the 'creates new' relation and the 'submits' relation, which both go to separate, but new entities. The 'creates new' relation goes to a 'PROFILE' entity, which was created to store all academic profile information. The weak 'submits' relation goes to the weak entity 'REPORT' which is meant to store any user-found grievances with information regarding the profiles. We removed any entities and relations that involved funding, in order to keep our goal realistic. We also renamed and reworked both the 'WORKS' and 'INTEREST' entities in the original diagram, into 'PUBLICATIONS' and 'TOPIC' respectively. Our 'EXTERNAL RESEARCHER' entity was replaced with a relation from 'PROFILE' to itself in order to maintain simplicity. The 'PROFESSOR' and 'STUDENT' entities were condensed into 'PROFILE' because it made our goal more concrete.

4 Implementation

4.1 RM Diagram



4.2 Significant Decisions in Model Conversion

There were no significant or unusual decisions made when converting our EER diagram to an RM diagram.

4.3 RM Changes During Implementation

Our 'Edit History' relation was not implemented in our system.

4.4 DBMS Implementation Decisions

The development of this project was undertaken in the Microsoft Windows environment. AppServ version 8.6.0 was used to facilitate the use of php, MySQL and phpMyAdmin [11]. At the time of writing, there is no cross platform functionality of this project as AppServ is not available for UNIX based systems. Cross-platform programs such as XAMPP are available, however, developing in this SQL development environment was not pursued, as our textbook and the provided course materials utilize the MySQL syntax [12]. The current distribution of XAMPP natively uses MariaDB, while this project's database uses MySQL[13]. Moving forward with this project we would choose to utilize a more current SQL database implementation for continuity, openness, and security. The MySQL database we created was utilized using MyPHPAdmin and MySQL-Front [14][15].

4.5 SQL Queries

4.6 Transaction Collection

- **Login Page**

```
SELECT user_id, is_admin, is_mod
FROM user
WHERE username = '[myusername]' and password = '[mypassword]'
```

- **Administration Page**

1. **Retrieve User Information**

```
SELECT user.user_id, user.username, user.is_admin, user.is_mod, user.start_date
FROM user
```

2. **Reverse Moderator Status**

```
SET is_mod=NOT is_mod
WHERE user_id=''
```

3. **Reverse Administrator Status**

```
SET is_admin=NOT is_admin
WHERE user_id=''
```

- **Moderation Page**

1. **Retrieve All Reports**

```
SELECT *
FROM report
```

2. **Mark a Report as Resolved**

```
UPDATE report
SET date_resolved=CURRENT_TIMESTAMP
WHERE report.profile_id=''
AND report.user_id=''
AND date_submit=''
```

- **Profile Page**

1. **Retrieve Profile Name**

```
SELECT name
FROM profile
WHERE profile_id=''
```

2. **Retrieve Location Information of a Profile**

```
SELECT i.i_name, i.postal_code, l.postal_code, l.country, l.city, p.I_postal
FROM institution i, location l, profile p
WHERE i.postal_code = l.postal_code
AND p.I_postal = i.postal_code
AND p.profile_id=''
```

3. **Retrieve the Topics of Interest of a profile**

```
SELECT interested_in.Tname
FROM interested_in, profile
WHERE interested_in.profile_id = profile.profile_id
AND profile.profile_id=''
```

4. **Retrieve Publications that the Profile has Authored**

```
SELECT authored.Pname, publication.link_to
FROM authored, profile, publication
WHERE authored.profile_id = profile.profile_id
AND publication.name = authored.Pname
AND profile.profile_id=''
```

5. **Retrieve Associated Profiles that the Profile has Worked With**

```
SELECT worked_with.profile_idB
FROM worked_with
WHERE profile_idA=''
```

- **Create Profile Page**

1. **Insert a New Profile**

```
INSERT INTO profile (user_id_creator, I_postal, name)
VALUES ([user_id], '[institution]', '[name]')
```

2. **Retrieve Profile ID**

```
SELECT profile_id
FROM profile
WHERE user_id_creator=[user_id]
AND I_postal='[institution]' AND name='[name]'
```

3. **Insert New Values into the "Interested in" Table**

```
INSERT INTO interested_in VALUES ([profile_id], '[topic name]')
```

4. **Insert New Values into the "worked With" Table**

```
INSERT INTO worked_with VALUES ([profile_idA], [profile_idB])
```

5. **Insert New Values into the "Publication" Table**

```
INSERT INTO publication VALUES ('[publication name]', '[link_to]')
```

6. **Insert New Values into the "Related to" table**

```
INSERT INTO related_to VALUES ('[topic]','[publication name]')
```

7. **Insert New Values into the "Authored" table**

```
INSERT INTO authored VALUES ('[profile_id]','[publication name]')
```

8. **Retrieve Institution Names and Postal Codes**

```
SELECT i_name, postal_code
FROM institution
ORDER BY i_name
```

9. **Retrieve Topic Names**

```
SELECT name
FROM topic
ORDER BY name
```

10. **Retrieve Profile Names**

```
SELECT name, profile_id
FROM profile
ORDER BY name
```

- **Report Profile Page**

1. **Add a New Report to the Database**

```
INSERT INTO report
VALUES ([profile_id], [user_id], NULL, CURRENT_TIMESTAMP,
'[report information]')
```

- **Search Page**

1. **Retrieve Profile id, Name, Institution, and Country**

```
SELECT profile.profile_id, profile.name, institution.i_name,
location.country, location.city
FROM profile
 JOIN location ON profile.I_postal=location.postal_code
   JOIN institution ON profile.I_postal=institution.postal_code
WHERE location.country LIKE '%[query]%'
```

2. **Retrieve Profile id, Name, Institution, and Country**

```
SELECT profile.profile_id, profile.name, institution.i_name,
location.country, location.city
FROM profile
 JOIN location ON profile.I_postal=location.postal_code
   JOIN institution ON profile.I_postal=institution.postal_code
WHERE location.country LIKE '%[query]%'
```

3. **Retrieve a List of Profile id's that are associated with a Topic**

```
SELECT interested_in.profile_id
FROM interested_in
WHERE tname='[topic]'
```

4. **Retrieve Topics Associated with a Profile id**

```
SELECT tname
FROM interested_in
WHERE profile_id=''
```

5. **Retrieve Profile id, Name, Institution, and Country**

```
SELECT profile.profile_id, profile.name, institution.i_name,
location.country, location.city
FROM profile
 JOIN location ON profile.I_postal=location.postal_code
   JOIN institution ON profile.I_postal=institution.postal_code
WHERE profile.profile_id=''
```

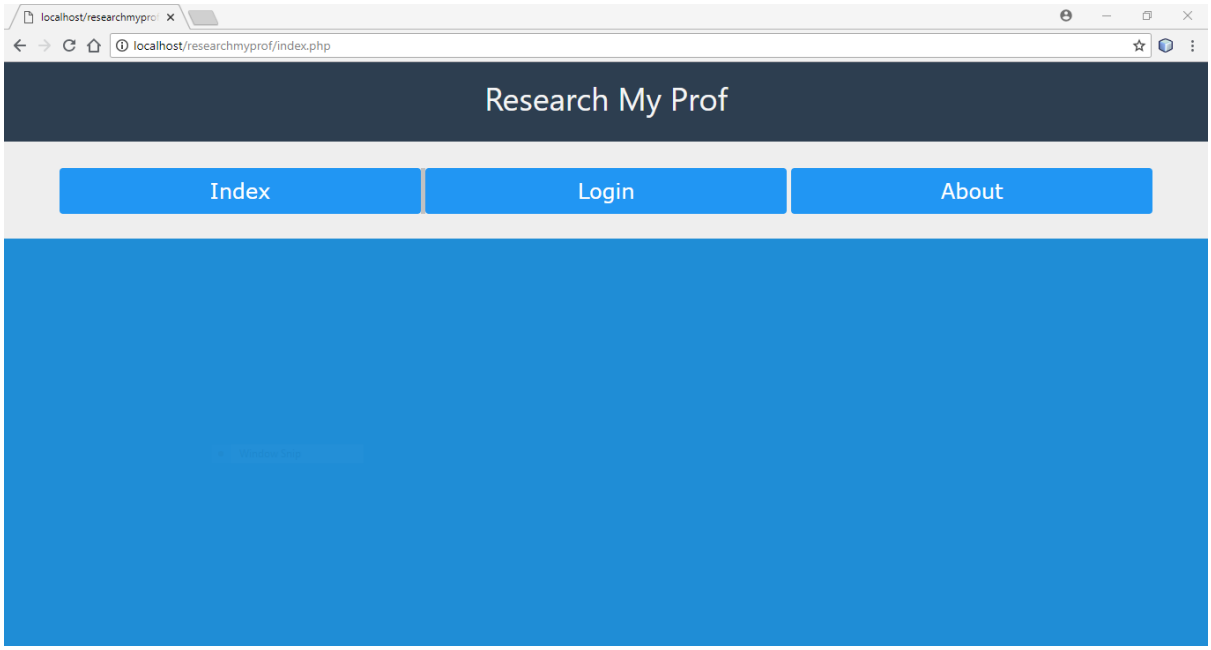
Our Session transaction has the following query:

```
SELECT username
FROM user
WHERE username = '[user_check]'
```

5 User Manual

5.1 General User

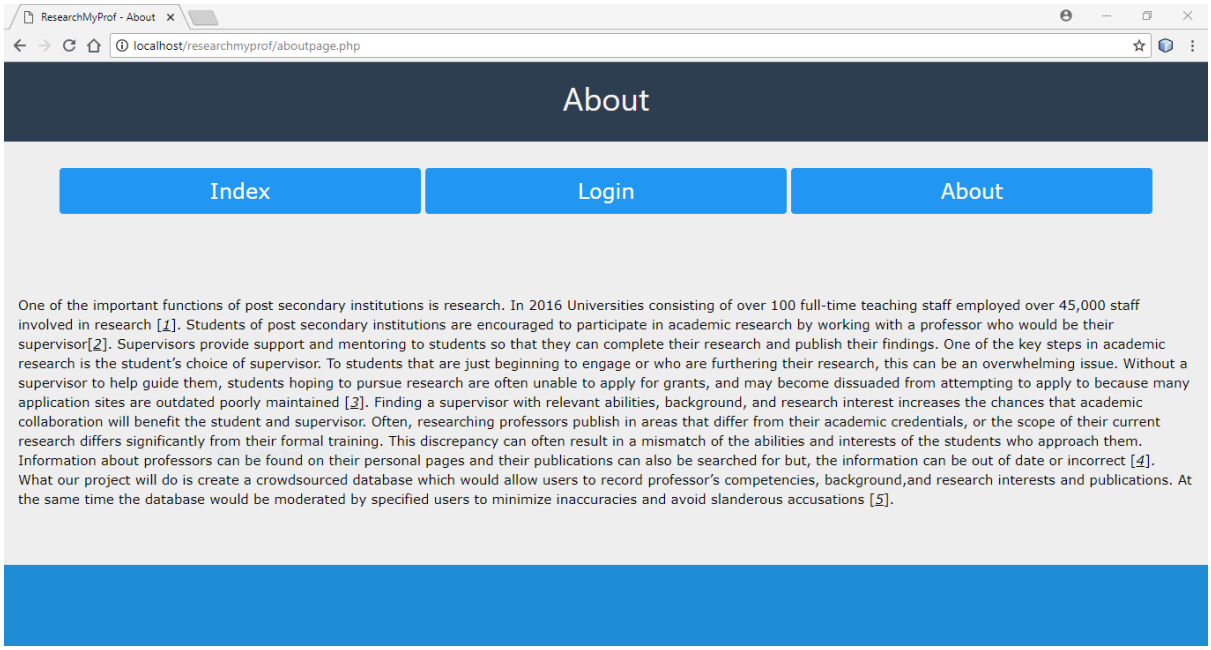
5.1.1 Main Index Page



The main index page generates a fixed location navigation bar which allows users to redirect to the main index page, login page, and an about page. From the navigation bar the website user can:

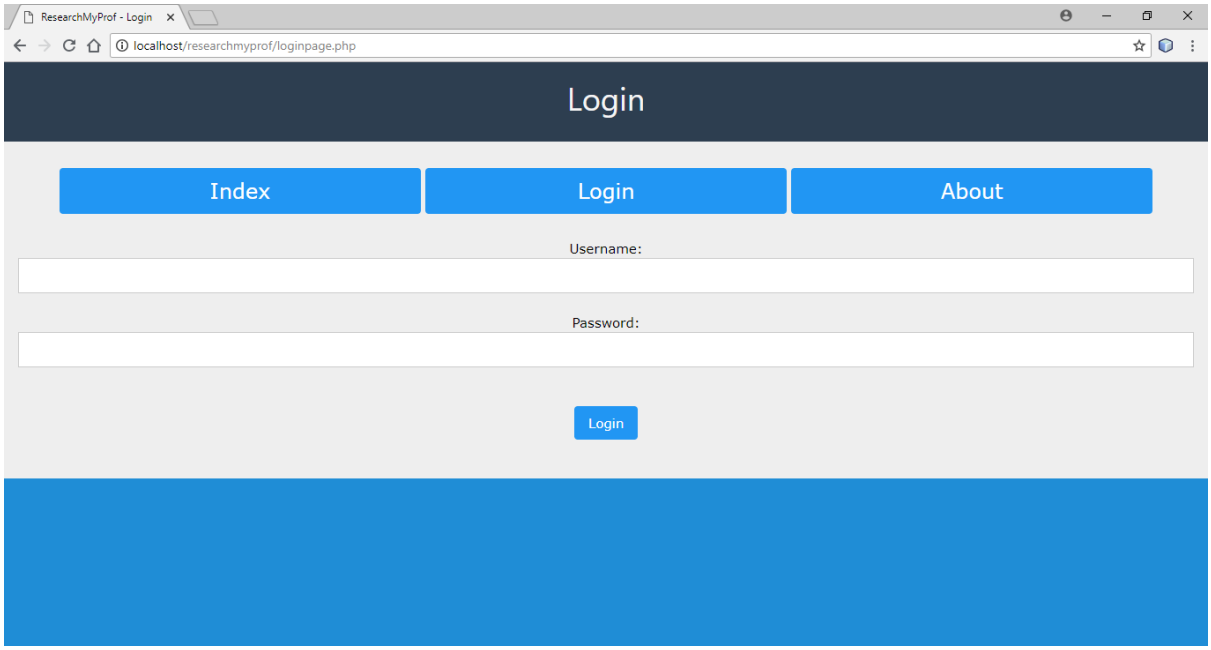
- Return to the main index page
- Login as one of three user types: User, Administrator, Moderator where Administrator and Moderator are overlapping subclasses of User.
- Read about the website and the motivation for its creation

5.1.2 About Page



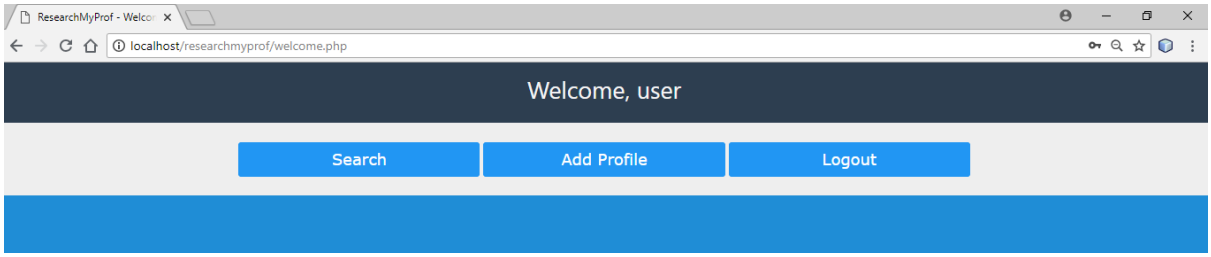
The about page contains contains the motivation for the creation of the website and a general explanation of the functionalities of the website. For further inquiry about the topics mention users may click the hyperlinked citations to be redirected to the externally hosted articles from which the basis of the project was founded.

5.1.3 Login Page



The login page allows user to enter their unique Username and a user defined password. Based off the information provided on this page, the user's account is queried from the database and they are redirected to the Welcome page. If the user's login information is not found in the database, they are informed that the information they provided is not valid, remaining on the login page with the ability to attempt to login again.

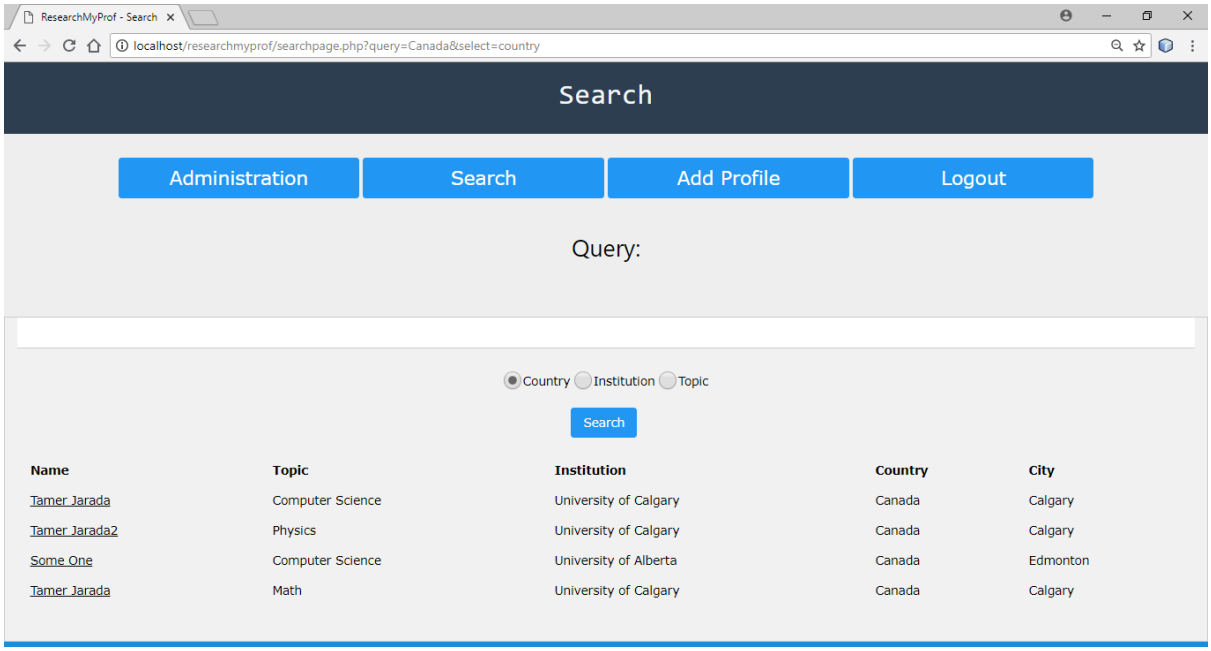
5.2 Welcome Page



The Welcome page appears when ever a User, Administrator or Moderator logs in which welcomes them by their Username. For all user types the Welcome page includes links the the Search, Add Profile and Logout Pages.

If the case that a User has Administrative and/or Moderator privileges, additional navigation options will be shown.

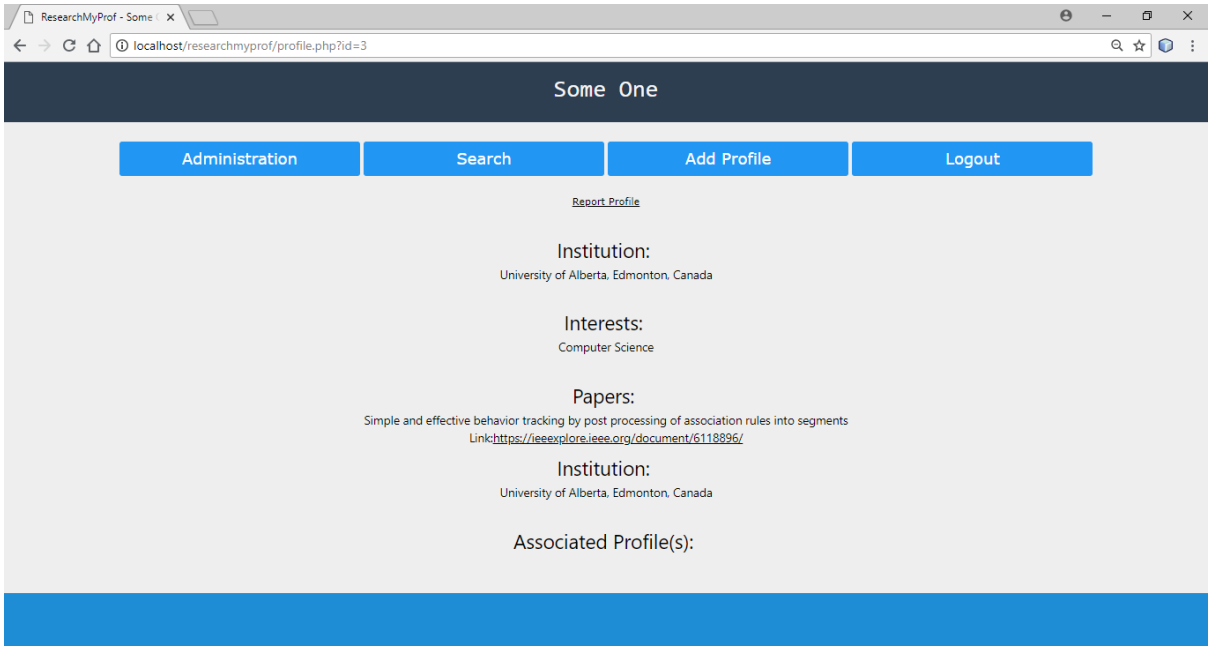
5.3 Search Page



The Search page available to all Users, allows for queries to be selected from the database. The search functionality allows users to enter query names which are not required to be case sensitive and supports sub-string searches of entries. There are three attributes that the User may filter search results by:

1. **Country**, that a profile's institution of research is found in. As a research may conduct research in several countries.
2. **Institution**, the institution where a research is currently researching at or where they have previously done research.
3. **Topic**, that is an area of research interest to the researcher.

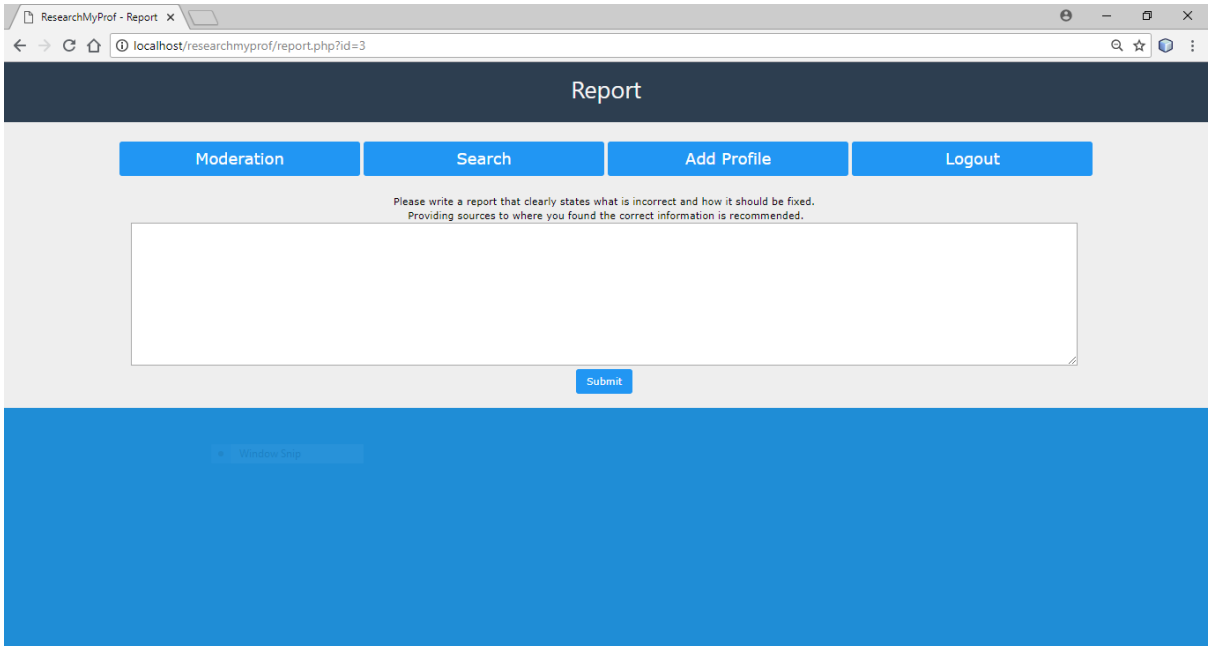
5.3.1 Profile Page



The profile page contains all the information that a general user would be in search of. It lists the institution of the selected profile, any interests that they are associated with, any works (including a link to the work, if applicable of course) and any other profiles they are associated with. Associated profiles are essentially meant to provide more information to the user about the profile in question. Associated profiles appear as a link to that profile where they can view more information, this is especially useful when the user is trying to narrow down group of profiles by interest, but in a more specific fashion.

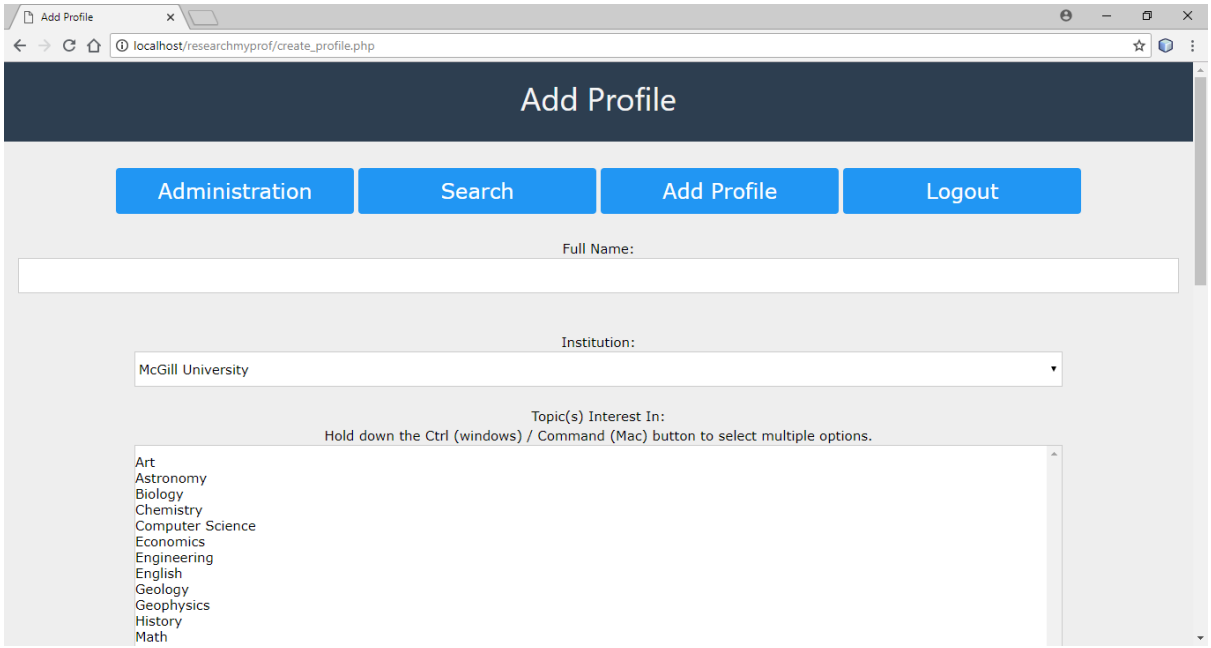
Right below the navigation buttons at the top of the page, there is a "Report Profile" button, which appears on all profile pages, it is used to ensure the information stored on a profile page is up to date and correct.

5.3.2 Report Profile Page



The Report Profile page can be accessed by any user through clicking "report profile" when viewing a profile. The page allows users to write a reason for their report which helps the moderator understand the reason for the report and can assist in rectifying the problem.

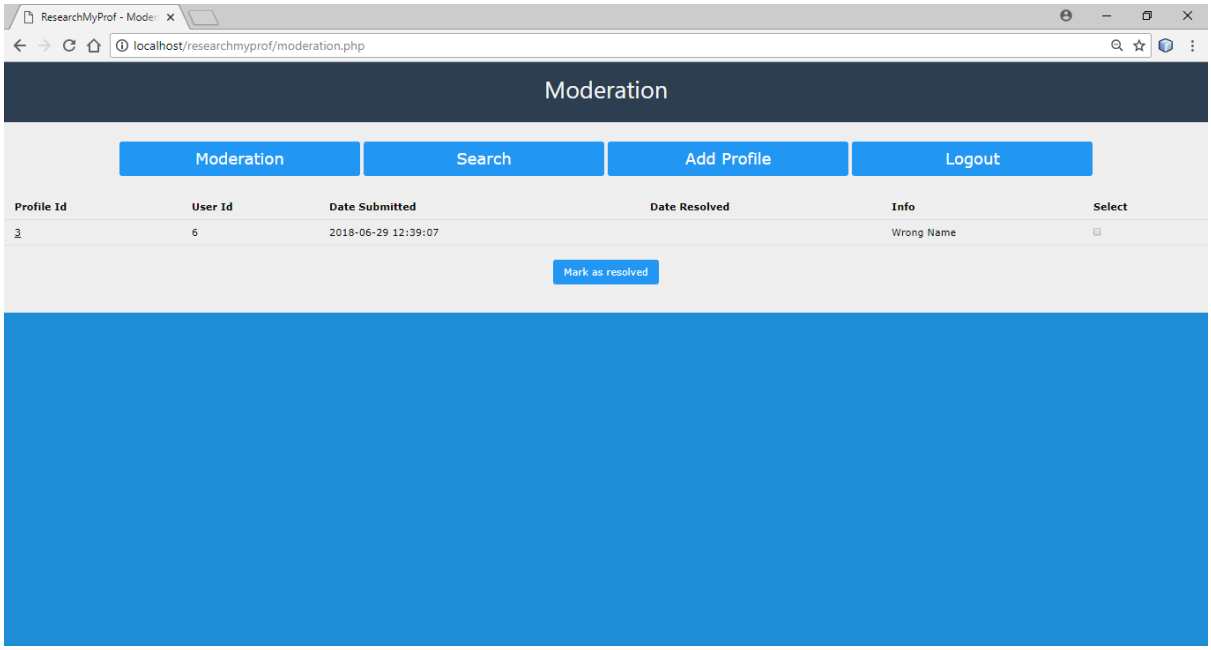
5.3.3 Create Profile Page



The Create Profile page allows the user to create a new profile for themselves or another researcher. To create a profile for a researcher the user must provide:

- **Full Name** in UTF-8 encoding of up to 255 characters
- **Institution** that the research is conducted at from a drop down menu of currently supported institutions.
- **Topics(s)** that the researcher is interested in from a provided list of institutions. The topics are provided to restrict the addition of non-academic topics of interested. The user may select one or more of the provided topics.
- **Publications Authored** allows the user to enter multiple publications associated with the researcher with a fixed format of: Text,Link,Topic Related to.
- **Associated Profiles**, which allows users to click to select one or more researchers whom already profiles.

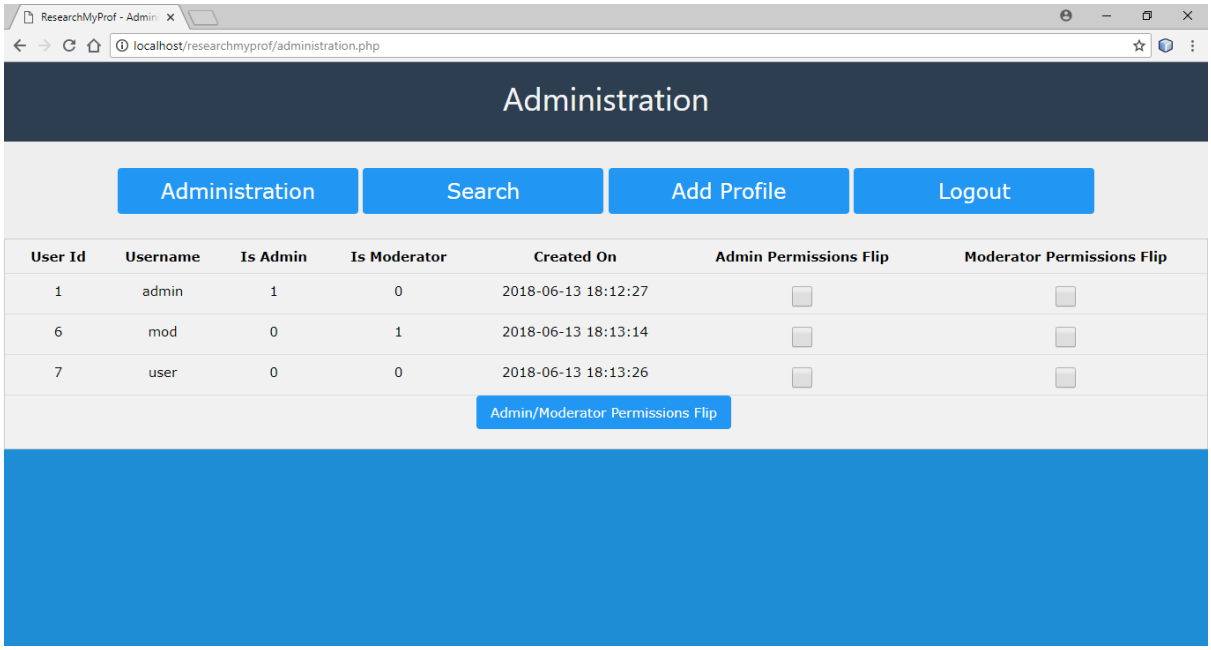
5.4 Moderator



Users who are moderators have access to the Moderation page. On this page Moderators can view the reports submitted by users. Each report submitted has a Profile id which refers to the the profile that has been reported for some reason. The Moderator may also see the User id of the user who submitted the profile. Additionally the dates for when the report was generated and the date for when the Moderator resolved the report are both presented. The information submitted by the reporter is also visible to ensure the moderator can properly understand the issue and ensure it is resolved.

Once the report is reviewed and dealt with, the moderator clicks the check box and then clicks the submission button labelled "Mark as resolved". At this time the resolved report is removed from view, but retained in the case that it must be examined at another time.

5.5 Administrator



The Administrator page contains a table of all Users in the database, user names, a binary flag for if the User is an Administrator (1=Yes, 0=No),a binary flag for if the User is a Moderator (1=Yes, 0=No), and the date their profile was created.

To change the privileges of a User in the data base, the Administrator clicks the box corresponding to the required change(s) and then clicks the submission button labelled "Admin/Moderator Permission Flip". For simplicity reasons the current database implementation allows Administrators to alter their own permissions, to allow for Administrators to surrender their positions or take on moderator roles in the event that the number of reports to resolve is too large for the number of moderators.

Appendices

Sample Database

Database: researchmyprof

Table: authored

profile_id	Pname
1	Simple and effective behavior tracking by post processing of association rules into segments
32	Cool Stuff
33	

Table: edit_history

profile_id	user_id	date_edited	edit_reason

Table: institution

postal_code	i_name
CB21TN	University of Cambridge
H3A0G4	McGill University
OX12JD	University of Oxford
T1K3M4	University of Lethbridge
T2N1N4	University of Calgary
T4N5H5	Red Deer College
T6G2R3	University of Alberta

Table: interested_in

profile_id	Tname
1	Computer Science
1	Math
32	Computer Science
33	Computer Science

Table: location

postal_code	country	city
H3A0G4	Canada	Montreal
T4N5H5	Canada	Red Deer
T6G2R3	Canada	Edmonton
T2N1N4	Canada	Calgary
CB21TN	Britain	Cambridge
T1K3M4	Canada	Lethbridge
OX12JD	Britain	Oxford

Table: profile

profile_id	user_id_creator	l_postal	name
1	1	T2N1N4	Tamer Jarada
32	7	T2N1N4	Kashfia Sailunaz
33	7	T2N1N4	Carey Williamson

Table: publication

name	link_to
Cool Stuff	www.coolstuff.com/coolstuffresearch.pdf
Simple and effective behavior tracking by post processing	https://ieeexplore.ieee.org/document/6118896/

Table: related_to

Tname	Pname
Computer Science	Simple and effective behavior tracking by post processing
Computer Science	Cool Stuff

Table: report

profile_id	user_id	date_resolved	**date_submit**	info
1	7	2018-06-18 21:45:59	**2018-06-17 19:32:07**	Its all wrong!
1	7	2018-06-25 23:04:59	**2018-06-18 21:45:59**	Test report

Table: topic

name
Art
Astronomy
Biology
Chemistry
Computer Science
Economics
Engineering
English
Geology
Geophysics
History
Math
Music
Neuroscience
Physics
Zoology

Table: user

user_id	username	start_date	password	is_admin	is_mod
1	admin	2018-06-13 18:12:27	admin	1	0
6	mod	2018-06-13 18:13:14	mod	0	1
7	user	2018-06-13 18:13:26	user	0	0

Table: worked_with

profile_idA	profile_idB
32	1
33	32
33	1

6 References

References

[1] Statistic Canada: http://www.statcan.gc.ca/tables-tableaux/sum-som/l01/cst01/educ68a-eng.htm

[2] Undergraduate Research - How To Start: http://www.ucalgary.ca/science/undergraduate/research/how_to_start

[3] Pure Award Applications: http://www.ucalgary.ca/provost/TLFC/awards/undergrad

[4] Usman Alim's CPSC page http://pages.cpsc.ucalgary.ca/~ualim/research.html

[5] Gender Bias in Rate My Professor Rankings: https://www.timeshighereducation.com/news/large-rate-my-professor-study-finds-gender-bias-comments

[6] Tamer Jarada's CPSC Page: http://pages.cpsc.ucalgary.ca/~tnjarada/

[7] Rate my Professor: http://www.ratemyprofessors.com/

[8] Research Gate URL https://www.researchgate.net

[9] W3 CSS Templates https://www.w3schools.com/w3css/

[10] Pure CSS Templates https://purecss.io/layouts/

[11] AppServe Homepage Site https://www.appserv.org/en/

[12] XAMPP Webpage https://www.apachefriends.org/index.html

[13] MariaDB Foundation page https://mariadb.org/

[14] PhpMyAdmin Homepage https://www.phpmyadmin.net/

[15] MySQLFront Homepage https://www.phpmyadmin.net/

[16] Imasri, Ramez and Navathe, Shamkant *Fundamentals of Database Systems 7th Edition*. Addison-Wesley Publishing Company, 2010.

[17] Referenced Report for Project layout http://sadatislam.me/pdf/BanglaBazarReport.pdf