Jarvis Huang
CS-GY 6083 – Project Part II
Due May 15, 2020

Web-Based Issue Tracking System Project

Project Introduction

This project is to build a web-based user interface for the issue tracking system with a back-end database, which involves interactions among users, projects, and issues. This report documents how the back-end database is designed and illustrates the functions of the front-end.

How to Run

1. Open terminal and move to the project folder
2. Run the program by entering command: php -S 127.0.0.1:8000
3. Open any standard web browser and enter the address 127.0.0.1:8000

Database Design

E-R Diagram

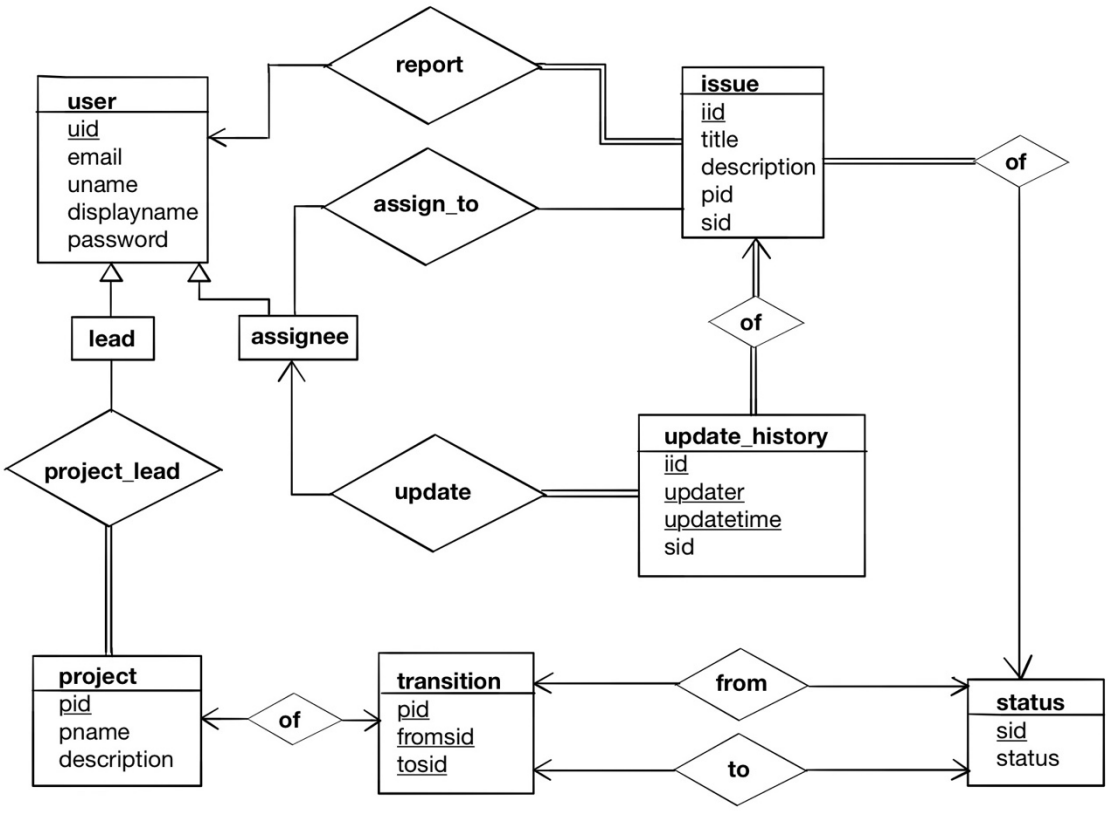


Diagram Explanations

1. `lead` and `assignee` are two overlapping subgroups of `user`.
2. All users are able to report an issue. Only assignees are able to update the issues assigned to them. Each update is made by only one assignee.
3. Each project has a unique workflow stored in the table `transition`.

Relational Schema

user (uid, email, uname, displayname, password)
project (pid, pname, description)
project_lead (lead, pid)
    (lead) is foreign key referencing `user` (`uid`)
    (pid) is foreign key referencing `project` (`pid`)
status (sid, status)
issue (iid, title, description, pid, sid)
    (pid) is foreign key referencing `project` (`pid`)
    (sid) is foreign key referencing `status` (`sid`)
assign_to (iid, assignee)
    (iid) is foreign key referencing `issue` (`iid`)
    (assignee) is foreign key referencing `user` (`uid`)
update_history (iid, updater, updatetime, sid)
    (iid) is foreign key referencing `issue` (`iid`)
    (updater) is foreign key referencing `user` (`uid`)
    (sid) is foreign key referencing `status` (`sid`)
transition (pid, fromsid, tosid)
    (pid) is foreign key referencing `project` (`pid`)
    (fromsid) is foreign key referencing `status` (`sid `)
    (tosid) is foreign key referencing `status` (`sid `)

Design Decisions and Assumptions

1. Users, projects, and issues are identified by their unique ID, which allows duplicate names, titles, or descriptions.
2. All IDs are auto-increment by MySQL. Therefore, new entries can put NULL at ID (`uid`, `pid`, `sid`, `iid`).
3. There is a separate table for each one of many-to-many relationships (users and projects, issue and assignees).
4. The `updater` in the `update_history` table must be one of the assignees for the issue. The only exception is the reporter. That is, even though a reporter may not be an assignee for the issue he reports, the system will create an entry in `update_history` with the reporter's ID and status "OPEN". To find reporter for an issue, the system will look for the first `update_history` entry for the issue. Therefore, the reporter is not listed in the `issue` table.
5. Leads of different projects are allowed to create statuses with the same name. All

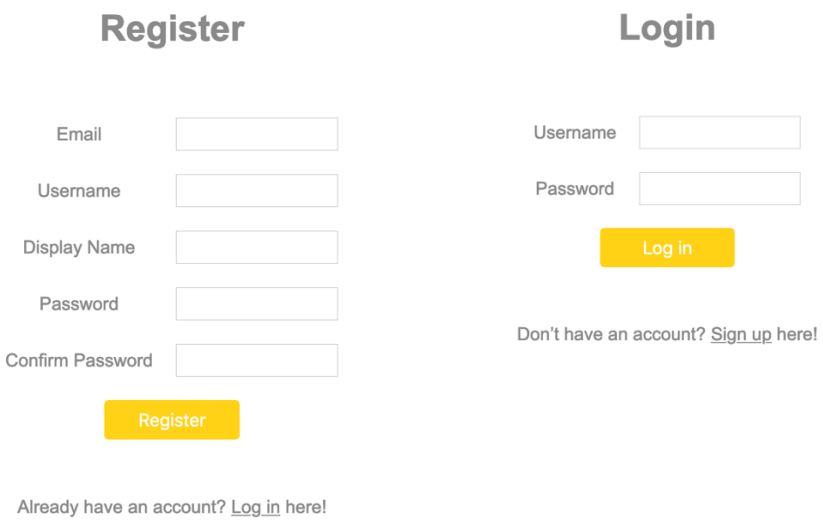
statuses are stored in a separate table `status`. Each status is identified only by its ID `sid`.

6. Different projects have different workflows, and one project has only one workflow. The table `transition` has all the workflows for all projects, and project ID `pid` is also a primary key for this relation.

Web Design Overview

The front-end web page is using PHP to connect the back-end database.

1. Register/Login



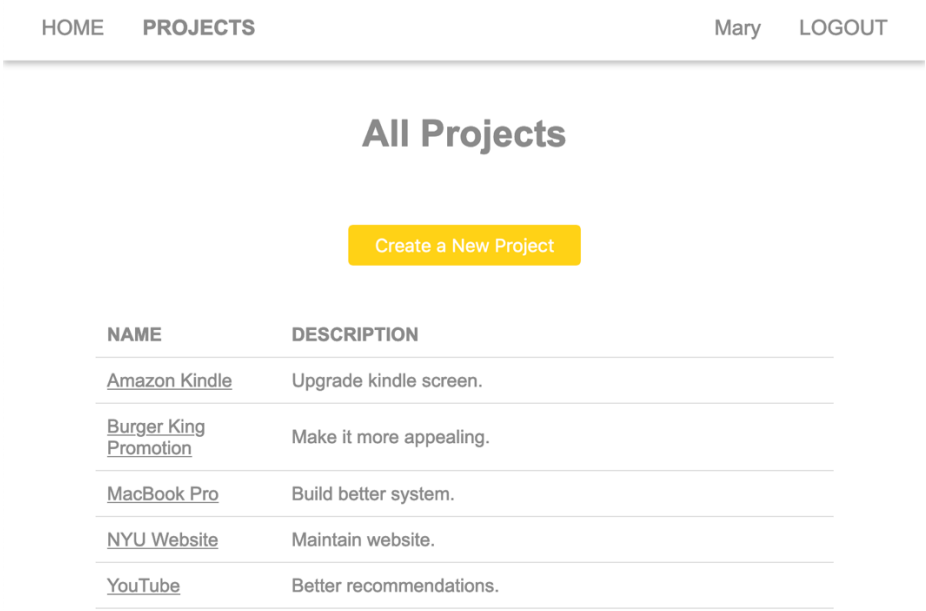
Users are free to switch between Register and Login page. One can create a new user account by entering email, password, username, and display name. Email and username for each account should be unique. If any information is not valid, the web system will show error informing user to re-enter information. Password encryption will be introduced in later section.

2. Logout

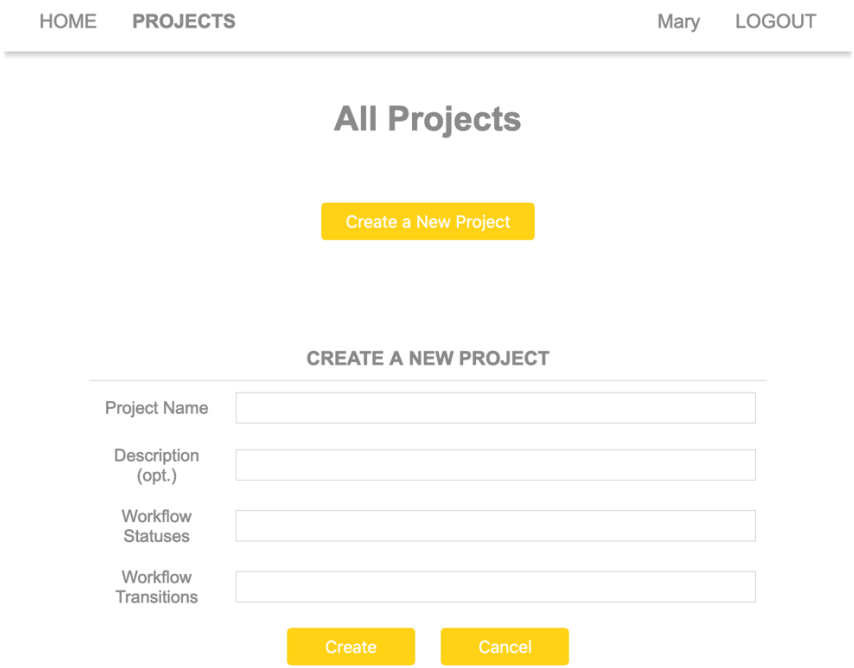
Once a user has logged in, there will be a nav bar above the page. The user's display name will be shown on the right top corner, with the Logout button next to it. After clicking the Logout button, the system will destroy the current session and log user out.

3. Projects

Every user who has logged in has access to see all projects in the Project page, as follows:



All users have the permission to create a new project by clicking the “Create a New Project” button, as follows:

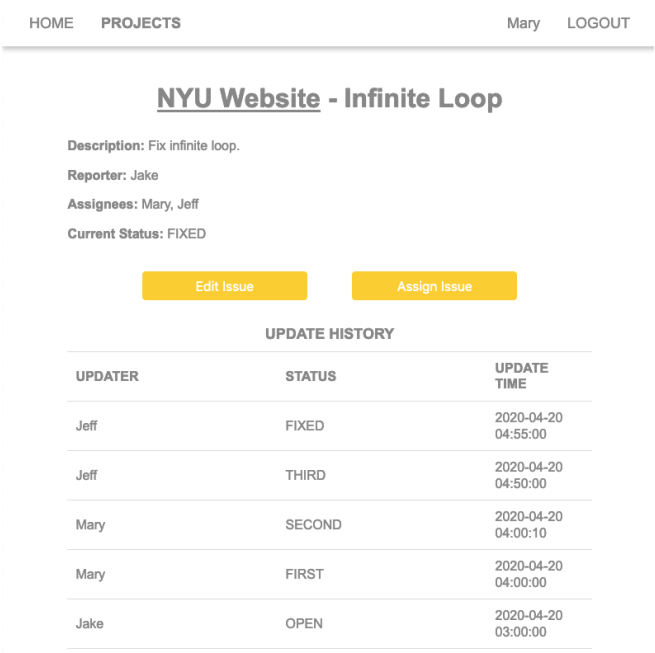


When a user creates a project, he/she should provide project name and description, enter the possible statuses, and initialize the workflow.

4. Project Dashboard

After the user clicked on a project in Projects page, the user could be redirected to the project dashboard, which shows the title, description, leads, and workflow transitions of this project, and lists all issues under this project. There is also a button for adding a project lead and one for reporting a new issue. Permission check will be discussed in later section. There is also a search bar and filter function. By entering one or some of the filter boxes, user is able to filter the issue and get desired results.

5. Issue Dashboard



After the user clicked on an issue in a project, the user could be redirected to the issue dashboard, which shows the project name, issue title, description, reporter, assignees, and current status of this issue, and lists all issues update history. Each update record shows the updater, updated status, and the timestamp. There is also a button for editing the issue and one for assigning the issue to a user. Permission check will be discussed in later section.

Functions Detail

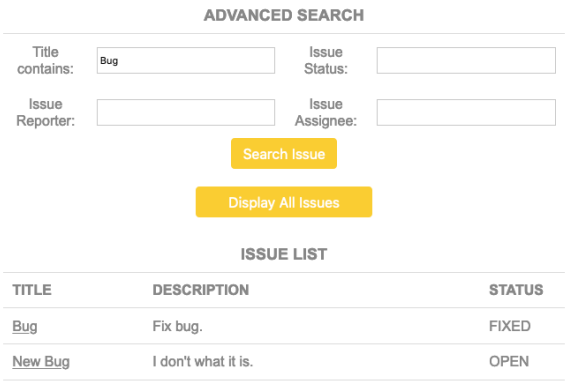
1. Filter and Search of Issue

   In the advanced search section in project dashboard page, there are four text boxes for user to enter: Title Contains, Issue Status, Issue Reporter, and Issue Assignee. The user is free enter one or more boxes to search for combinations of a few conditions. For example, follows are all issue lists:

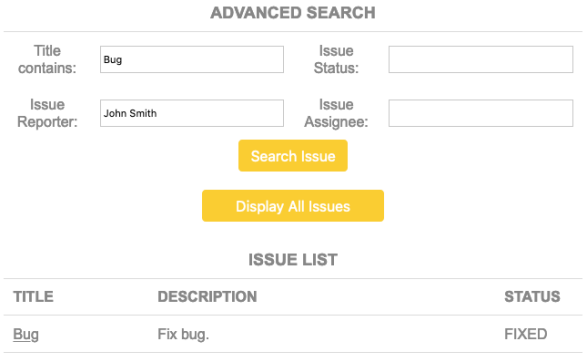
ISSUE LIST

TITLE	DESCRIPTION	STATUS
Infinite Loop	Fix infinite loop.	FIXED
Bug	Fix bug.	FIXED
New Bug	I don't what it is.	OPEN

   If the user wants to find all issues whose title contains word “Bug”, the user can enter “Bug” in Title Contains field and ignore other fields and then click Search Issue button. The result is as follows:



Another example, if user enter more than one filter boxes.

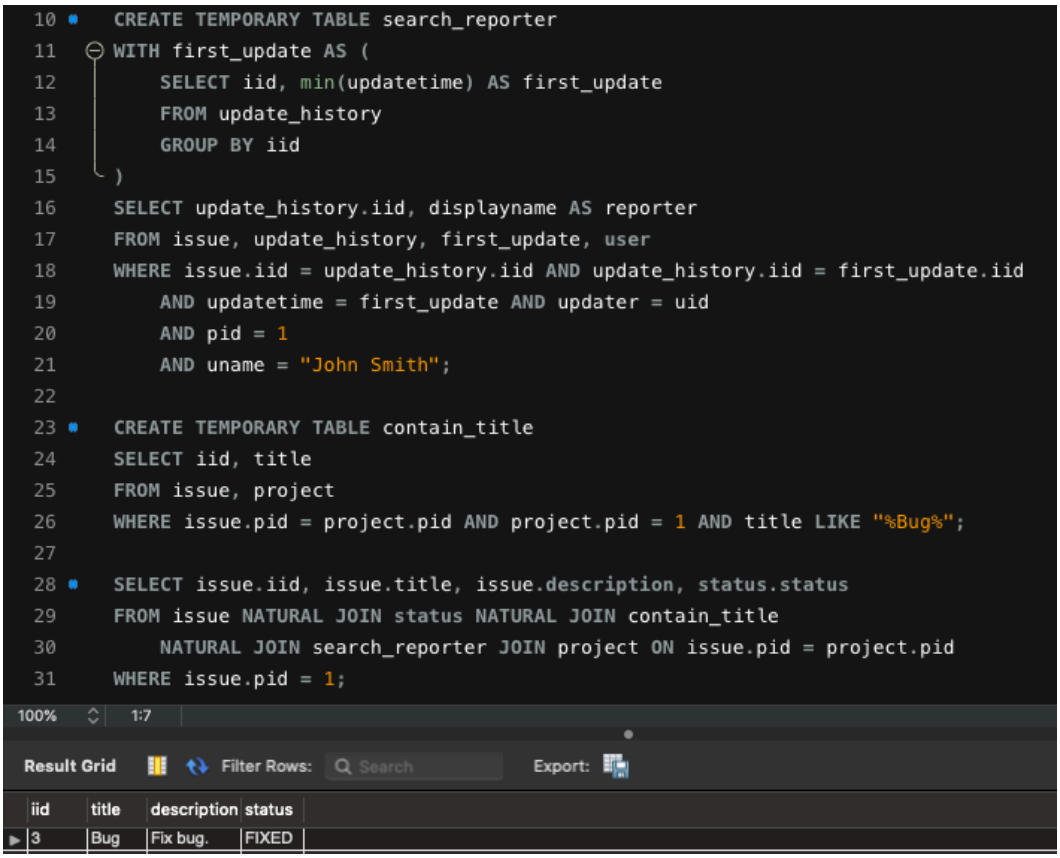


The actual SQL query will be as follows:

```
CREATE TEMPORARY TABLE search_reporter
WITH first_update AS (
    SELECT iid, min(updatetime) AS first_update
    FROM update_history
    GROUP BY iid
)
SELECT update_history.iid, displayname AS reporter
FROM issue, update_history, first_update, user
WHERE issue.iid = update_history.iid AND update_history.iid = first_update.iid
    AND updatetime = first_update AND updater = uid
    AND pid = 1
    AND uname = "John Smith";

CREATE TEMPORARY TABLE contain_title
SELECT iid, title
FROM issue, project
WHERE issue.pid = project.pid AND project.pid = 1 AND title LIKE "%Bug%";

SELECT issue.iid, issue.title, issue.description, status.status
```

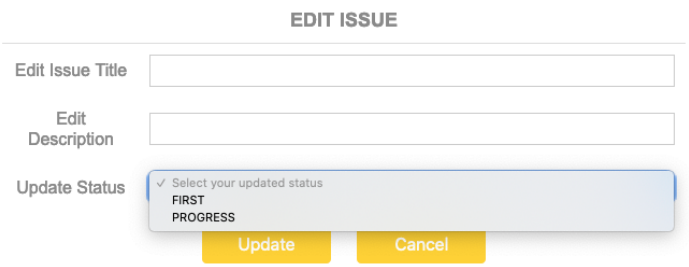
```
FROM issue NATURAL JOIN status NATURAL JOIN contain_title
    NATURAL JOIN search_reporter JOIN project ON issue.pid = project.pid
    WHERE issue.pid = 1;

DROP TEMPORARY TABLE search_reporter;

DROP TEMPORARY TABLE contain_title;
```

There is screenshot of running the query above in MySQL:



To further explain this function, we can see how back-end realize it. Each filter text box will create a temporary table with corresponding results. Finally, the system run a query by joining all temporary tables and get combination results. The system will drop all temporary tables after one advanced search.

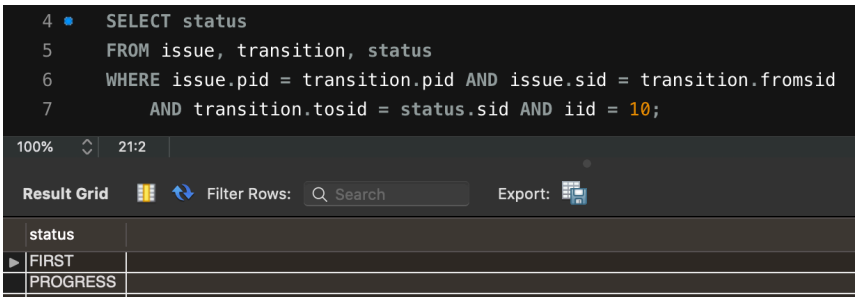
2. Issue Status Update



The project lead, issue reporter, and issue assignees are allowed to edit issues. User is able to enter one or more fields to edit the issue, but only the update of status with the edit time will be stored in database. As the picture shows, Update Status is a drop-down menu which lists all possible next statuses according to the current status of the issue. The following query finds all next possible statuses:

```
SELECT status
FROM issue, transition, status
WHERE issue.pid = transition.pid AND issue.sid = transition.fromsid
    AND transition.tosid = status.sid AND iid = 10;
```

The result shows as follows:



Permission Check

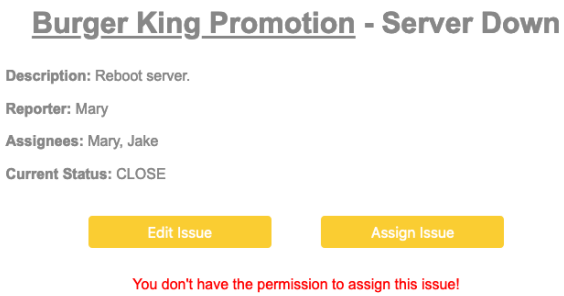
As the project designed, all logged in user have access to see all projects. The user who has created the project will automatically become the first lead of the created project. Any user has the permission to reporter an issue to any project. The issue creator will automatically become the only reporter of this issue.

The permission check in this project includes:

1. Only the existing leads have the permission to add other users as leads of the project.
2. Only the existing leads have the permission to assign an issue in his/her project to a couple of users, who will become the “assignees” of this issue.
3. Only the existing leads, issue reporter, and issue assignees have the permission to edit the issue.

If the current user failed the permission check, the web system will show a message notify

the user as follows:



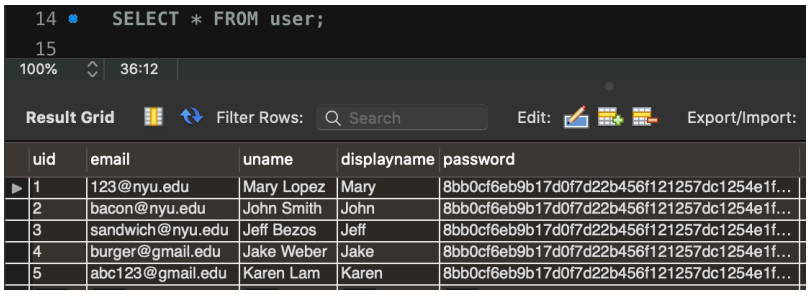
Note that Mary is the reporter and assignee of the issue, but she is not a project lead (project leads information shows in project dashboard page but does not show in the issue dashboard page). Therefore, she is not allowed to assign the issue, but she is allowed to edit the issue.

Password Encryption

The project uses SHA-256 to encrypt password. SHA-256 is cryptographic hash which generates a unique 256-bit signature for a text. It is a one-way cryptographic function and the result is of fixed size for any size of source text. The signature cannot be decrypted back to the original text. Therefore, it ensures the security of the user private information.

When a user registers a new account, after validating all registration information, the system will encrypt the user password using the PHP function `$encrypt_password = hash("sha256", $register_password)`, and stored the encrypted password in the database.

When a user tries to login with a username and a password, the system will encrypt the password using SHA-256 and compare it with the one stored in the database to see if they match. Here is a screenshot of the `user` table stored in the database:



uid	email	uname	displayname	password
1	123@nyu.edu	Mary Lopez	Mary	8bb0cf6eb9b17d0f7d22b456f121257dc1254e1f...
2	bacon@nyu.edu	John Smith	John	8bb0cf6eb9b17d0f7d22b456f121257dc1254e1f...
3	sandwich@nyu.edu	Jeff Bezos	Jeff	8bb0cf6eb9b17d0f7d22b456f121257dc1254e1f...
4	burger@gmail.edu	Jake Weber	Jake	8bb0cf6eb9b17d0f7d22b456f121257dc1254e1f...
5	abc123@gmail.edu	Karen Lam	Karen	8bb0cf6eb9b17d0f7d22b456f121257dc1254e1f...

XSS/CSRF protection

All web pages, except Login and Register, will connect to the session and checks that the user has authenticated and that the remote IP address matches the address used to create a session. If the user is not authenticated, the system will log user out and redirect the user to the Login page. Here a snippet of code for XSS/CSRF protection:

```php
function check_session() {
    // Resume the existing session
    session_start();

    // Check if the user has logged in
    if (!isset($_SESSION["login_username"])) {
        // Redirect to the home åpage
        header("Location: ../app/login.php?login=no_session");
        exit;
    }

    // Check if the request is from a different IP address to previously
    if (!isset($_SESSION["loginIP"]) || ($_SESSION["loginIP"] != $_SERVER["REMOTE_ADDR"])) {
        header("Location: ../app/logout.php");
        exit;
    }
}
```

SQL injection prevention

All input fields in this project uses the PHP function real_escape_string to escapes special characters in the input string for use in an SQL statement, taking into account the current charset of the connection.

All information fetched from the back-end database uses the PHP function htmlspecialchars and stripslashes to convert special characters to HTML entities, protecting the HTML web page from being affected or hacked by the data from the back-end database.