**Brandon Vo (bdv9527@nyu.edu)**

**Emily Lee (el2679@nyu.edu)**

**Github Project Link:  https://github.com/bvo4/Databases-Project.git**

**Databases Project Part 2**

This project features the creation of a site built using a mixture of MySQL, HTML, PHP, and CSS. The purpose of this web site is to provide a questions and answers message board where users can access a list of questions and browse the list of answers posted for each question, if present.  The accuracy of such answers are identified by a rating system and a best answer system to differentiate strongly written responses from weak answers.

Each question is labeled with a title, body of text, and a topic under which that particular question belongs to.  As such, any answers written will be directed at a question and display a rating system based on the number of likes provided.  In addition, an answer can be selected by the question poster as the best answer to that question in which its grade will be greatly increased.

Every user with a basic set of profile information pertaining to their username, password, profile details, and location are stored with each user.  This information can be freely changed if specified by the owner of that user account.  Any user who is logged in will have their username recorded, allowing them to post any questions and answers and rate other answers if desired. In addition, all users will be identified as one of the three statuses:  Basic, Intermediate, and Expert.  These statuses are not changeable and can only be attained once a user has received a threshold of 20 likes or 50 likes for intermediate and expert status respectively.
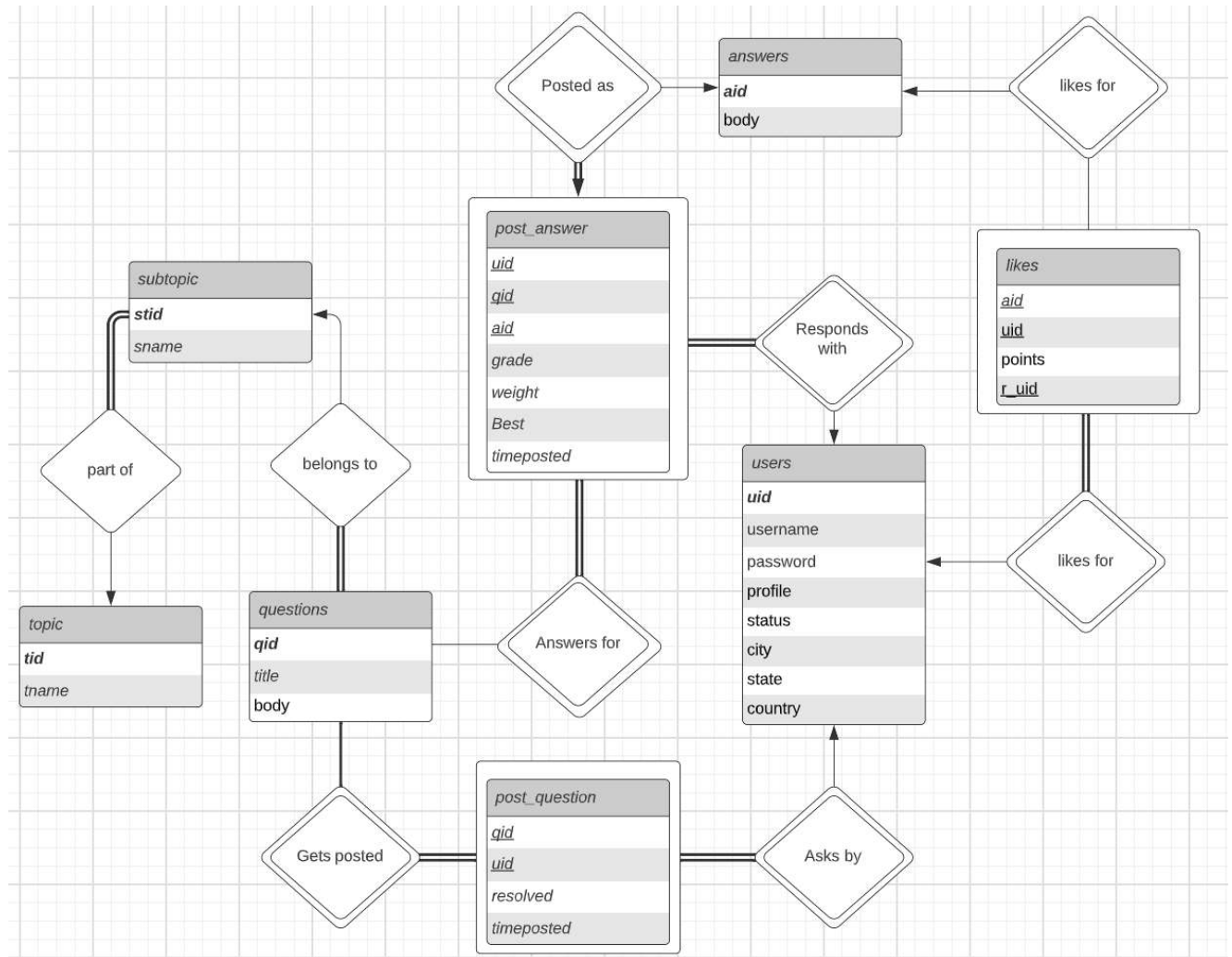
**MySQL tables being used:**

There are a total of 8 tables that were created for the purpose of supporting the underlying database of our Q&A web page.  The purpose of each table will be explained below.

**Users**:

The users table contains the list of all users who have signed up to participate for the website.  This table will contain details pertaining to the user's location, status, password, username, and profile.  All details, excluding status and uid, can be changed if the user wishes to do so to reflect a change in the user's identity.  This change will permeate to all posts made by this user and is purely for adding visual information.

Every user will begin with a status of "Basic."  This reflects the lowest form of status as the user will begin with an overall rating of 0 likes.  This rating system is not

contained within the user's table because the user's ratings will be reflected in a separate table to reduce redundancy.

**Likes:**

As mentioned in the users table, the likes table will store information pertaining to what answers have been liked by a user and the id of the user whose answer is being liked.  It will contain only the foreign key to the user id of the user who liked the answer, the foreign key pointing towards the answer id of the answer being liked, and the foreign key of the user whose answer is being liked.  In addition, this table will store a trigger that, upon the insert or deletion of any new like, will allow the table to check on the total number of likes a user has received and determine if the user has met or dropped below the threshold necessary to elevate or relinquish his/her status.

Every like will provide a rating of 1 point for the purpose of this web page, so while points are not used by the web page, it is being stored in the event that it has been deemed that an answer should receive more points by an administrator.

**Questions:**

The questions table will contain a list of all questions posted including the id, the title, body, and subtopic id.  The information kept in this table pertains to only the question itself rather than the circumstances that led to this question being posted which will be stored in another table.

The site will query for this table along with the post_questions, subtopic, and the topic table to gather all information related to a particular question.  Furthermore, the subtopic id will be a foreign key tied to the subtopic table which will contain a foreign key to the topic table.  This allows a question to be identified by a subtopic and main topic which are required for the creation of this question.

**Post_Question:**

The Post_Question table contains the metadata related to the circumstances that led to a question being posted.  This table contains the id of the question being posted, the user id of the user who posted the question, a boolean that indicates whether or not the question has been resolved or not, and a timestamp that indicates when the question was posted.

This table is intentionally made separate from questions in order to maintain concise data storage and prevent redundant design.  For such scenarios where the site would want to acquire data such as the total number of questions posted or an

abbreviated count of unresolved questions or finding the oldest question, it would be unnecessary to query for the question's contents or subtopics.

The resolved status of a question will be directly tied to whether or not a best answer has been chosen. If there is no best answer chosen, then the answer will be marked as unresolved.

**Topic:**

The Topic table contains all of the ids and names pertaining to the list of topics available to choose from. These topics are not directly tied to the posted questions, but are directly related to the subtopics that the questions are linked to.

This table is used primarily for advanced searches where the user might prefer to search for a broad selection of questions under a general topic's branch. This is also used for identifying a common attribute shared by multiple subtopics. The set of topics are provided by the administrators and cannot be created by a user.

**Subtopic:**

Similar to the topic table, the subtopics are used primarily to aid the user in search queries. Every question will be tied to a subtopic located in the subtopic table. Every subtopic will be linked to a larger main topic in the topic hierarchy to further aid in identifying the type of question being asked.

Similarly to the topic table, the set of subtopics are provided by the administrators and cannot be created by a user.

**Answers:**

The Answers table contains a list of all answers with their corresponding id and body (the actual answer). This information only consists of the answer itself, and other information such as when it was posted will be in the Post_Answers table which is directly linked to this table.

**Post_Answers:**

The Post_Answers table contains the information on which question the answer was posted to, when it was posted, and who posted it. Foreign key uid references the Users table, qid references the Questions table, and aid references the Answers and Likes table. The table also includes the answer's grade (based on likes), weight of each like (the likes weigh more if it is the best answer), where it is the best answer, and the time posted.

This table is used to store user interaction with the answers separately from the question, user, and the answer itself.  Similarly to Post_Question, this table was intentionally made separate from the Answers table in order to reduce redundancy in the event that we would need to check for an answer's metadata instead of the contents of the answer itself.

**Website Construction:**

For the construction of the website design, the webpages were constructed using a library set from Reactjs and Bootstrap.  Reactjs is a set of library components with additional CSS styling variations for aesthetic purposes.  However, ReactJS was provided with limited use because most of ReactJS's functionality is available only through Javascript functions, and as such, requires the web page's HTML function to be written in Javascript.  This would conflict with PHP's implementation as PHP is required to write the HTML outputs of its queries.  Relying too much on ReactJS would mean having to write the HTML web page on PHP on top of HTML which was considered too unreliable to use.  As a result, HTML was used only for CSS styling.

For convenience and quick access to most of the site's contents, every page is created with a header.  This header will provide a link to the homepage, the questions page, and a user tab.  If the user is not logged in, the user page will be identified by a generic dropdown written as "User Page."  If the user is logged in, then the dropdown will display the user's username instead.This menu will provide a link to login to the site, access to the user's profile page and a sign out option.

**Homepage**



| Username: | Topic: | Title: | Body: | Date: | View Answers: |
|---|---|---|---|---|---|
| Mike6 | Other | HOW DO I FLY | I try flapping my arms, but it doesn't work. | 2020-04-22 20:23:59 | View More |
| Daryl9 | Microeconomics | HOW TO PAY TAXES | Help, the government is inquiring about my car's extended warranty | 2017-02-07 05:31:30 | View More |
| Janet1 | Botany | HOW TO GROW PLANT | My crops keep dying when I feed it coca-cola. What am I doing wrong. | 2016-12-12 13:41:17 | View More |
| Mike6 | Algebra 1 | HOW TO 1+1 | HELP, HOW DO I COMMIT 1+1 | 2013-03-12 01:14:56 | View More |
| Daryl9 | Cryptocurrency | Bitcoin vs. Ethereum | What is the difference between Bitcoin and Ethereum? Which one should I buy into? | 2012-11-01 06:25:00 | View More |

The user is first presented with the homepage.  The homepage provides quick navigation to the rest of the site where the header features all links available to a

non-user which includes a link to a list of all questions, a browsing page to access the search functionality, and an option to login or register an account.

The homepage will automatically make a query to list the five most recently posted questions on the page along with an option to view any answers for each of them.

**select** *
**from** questions, post_question, users, subtopic
**where** questions.qid = post_question.qid
**and** post_question.uid = users.uid
**and** questions.stid = subtopic.stid
**order by** timeposted desc
**limit** 5

## Questions Page

When the questions page is loaded by default, it will query for the entire list of questions in descending order of chronological time. The following query was used to acquire the list of questions:

**select** *
**from** questions, post_question, users
**where** questions.qid = post_question.qid
**and** post_question.uid = users.uid
**order by** timeposted desc;

Home   Questions  Browse  User Page ▼

### QUESTIONS PAGE

| Username: | Topic: | Title: | Body: | Date: | View Answers: |
|---|---|---|---|---|---|
| Mike6 | Other | HOW DO I FLY | I try flapping my arms, but it doesn't work. | 2020-04-22 20:23:59 | View More |
| Daryl9 | Microeconomics | HOW TO PAY TAXES | Help, the government is inquiring about my car's extended warranty | 2017-02-07 05:31:30 | View More |
| Janet1 | Botany | HOW TO GROW PLANT | My crops keep dying when I feed it coca-cola. What am I doing wrong. | 2016-12-12 13:41:17 | View More |
| Mike6 | Algebra 1 | HOW TO 1+1 | HELP, HOW DO I COMMIT 1+1 | 2013-03-12 01:14:56 | View More |
| Daryl9 | Cryptocurrency | Bitcoin vs. Ethereum | What is the difference between Bitcoin and Ethereum? Which one should I buy into? | 2012-11-01 06:25:00 | View More |
| Leffen | Algebra 1 | Difference between adding and multiplication | What's the difference between adding 1+1+1 and 1*3? | 2012-03-14 17:24:23 | View More |
| Bobby2 | CyberSecurity | HOW TO HACK | Hackerman keeps demanding ransom in bittycoin. How do I pay. | 2011-11-30 18:56:23 | View More |
| Mike6 | CyberSecurity | Best antivirus program | What antivirus program should I download for my computer? Any recommendations? | 2009-08-07 04:59:16 | View More |
| Elphelt | Other | Kingdoms in microbiology? | What are the 3 domains of life that live in the microbiology world? | 2007-12-15 03:54:52 | View More |
| Bobby2 | Other | Difference between loops? | What's the difference between a for loop, a while loop, and a do-while loop? | 2001-06-13 11:44:51 | View More |

The questions are posted in reverse chronological order. Any questions that are marked as resolved will have their entries recolored in pink while unresolved questions remain as the default color. Every question will have the username of the author of the

question, the title of the question,the body of the question, the subtopic of the question, and the date the question was posted on as relevant information pertaining to the question.

Every question will also provide an option to view answers made to that question, if there were any answers posted. This will lead to a separate page where the qid of the question will be send to the answers.php page, allowing the web page to make a query for the list of answers directed at that particular question.

### Submitting a Question

If the user is logged in while looking through questions, he/she will be presented with an option to submit a question of their own. If this option is chosen, then the user will be sent to a screen asking him/her to submit a form inputting the question title, the question body, and one of the many pre-selected topics. The page will send a query to the subtopic table in order to get the entire list of subtopics to choose from, and the topic table isn't chosen because every subtopic is created as part of a larger main topic. Therefore, when a user chooses a sub-topic, that user will be declaring that question to be a part of a larger main topic in the topic hierarchy.

When the user submits his/her question, the web page will send two queries to the questions table to identify the contents of the message benign asked and to the post_questions table to identify who is asking the question and the status of this question. The query to the questions table will create a new entry containing the new qid, the subtopic's id, the title of the question, and the body of the question in the following query:

**INSERT INTO** questions(qid, stid, title, body)
**VALUES**($qid, $_POST[stid], '$_POST[title]', '$_POST[body]

The second query is sent to post_questions where the web page will use the qid inputted in questions. The web page will also record the current timestamp of when the post was submitted and input it alongside the userid, and the web page will automatically set the question as unresolved.

"INSERT INTO post_question(qid, uid, resolved, timeposted)
VALUES($qid, $_SESSION[uid], False, '$date')";

## Submit Question Page

| Question Title | |
| --- | --- |

| Question Body | |
| --- | --- |

Select a topic:

Select a topic:

Databases

Submit

---

**Answers Page**

When "View More" is clicked on a specific question, the qid of that question is stored and sent to the answers.php page. The answers.php page will run the following query:

**select** *
**from** answers, post_answers, users
**where** post_answers.qid = $qid
**and** answers.aid = post_answers.aid
**and** users.uid = post_answers.uid
**order by** grade **desc,** timeposted **desc**;

Where $qid represents the stored question id from the questions.php page. The list of answers will be displayed in an order where the best answer will always be shown first, then the answers will be listed based on how many likes they have received, then finally it will be sorted in reverse chronological order.

### ANSWERS PAGE

**HOW TO 1+1**

HELP, HOW DO I COMMIT 1+1

Submit Answer

| Body: | Username: | Likes: | Date: | Leave a Like? | Best Answer |
| --- | --- | --- | --- | --- | --- |
| If you jump underwater, you're technically you're flying because you push water down like birds push air down. | Darrin4 | 6 | 2023-12-31 16:11:09 | Like | This is Best Answer |
| Git add git commit. | Daryl9 | 20 | 2019-11-11 23:55:29 | Like | |
| Workout by doing jumping jacks. Maybe you'll jump high enough to fly. | Mang0 | 1 | 2021-07-22 03:41:18 | Like | |
| Have you tried riding an airplane? | JV4 | 0 | 2021-07-22 01:16:53 | This is your answer! | |
| You cannot fly. We are bipeds, not birds. | JV4 | 0 | 2013-04-23 16:11:09 | This is your answer! | |

A demonstration of our answers page. The table row is marked in red to indicate that this has been marked as the best answer.

## Submitting Answers

Similar to questions, answers will provide an option to submit an answer if the user is logged in. If selected, the user will be taken to a page that will provide him with the details of the question the user is trying to answer and provide a textbox for the user to type his/her answer.

The answers page sends the qid of the corresponding question to post_answers.php. The page will make a query to the MySQL server to obtain the question title and body to inform the user of the details regarding the question. Once the user types his/her answer and submits, the page will make two queries. One query will create a new answer in the answer table containing the next aid and body.

"**INSERT INTO** answers(aid, body)
**VALUES** ($aid, '$_POST[body]')";

The second query will submit the post to post_answers to remember who posted the answer. In this table, the web page will make a new insert based on the qid of the question being answered, the user's id, the new aid for this answer, and the current time and date when this answer was submitted.

"**INSERT INTO** post_answers(uid, qid, aid, grade, weight, best, timeposted)
**VALUES** ($_SESSION[uid], $qid[qid], $aid, 1, 1, False, '$date')";

## Submit Answer Page

| Question Title: | HOW DO I FLY |
| Question Body: | I try flapping my arms, but it doesn't work. |
| Answer Body: | |

Submit

**Like & Best Answer functionality**

HOW TO 1+1

HELP, HOW DO I COMMIT 1+1

Submit Answer

| Body: | Username: | Likes: | Date: | Leave a Like? | Best Answer |
|---|---|---|---|---|---|
| If you jump underwater, you're technically you're flying because you push water down like birds push air down. | Darrin4 | 6 | 2023-12-31 16:11:09 | Unlike | This is Best Answer |
| Workout by doing jumping jacks. Maybe you'll jump high enough to fly. | Mang0 | 1 | 2021-07-22 03:41:18 | Like | |
| Have you tried riding an airplane? | JV4 | 0 | 2021-07-22 01:16:53 | Like | |
| Git add git commit. | Daryl9 | 20 | 2019-11-11 23:55:29 | Like | |
| You cannot fly. We are bipeds, not birds. | JV4 | 0 | 2013-04-23 16:11:09 | Like | |

Two images above show the different states of the like system. Any user who views their own question will be informed about their own answer and will be unable to like any posts made by the same user. Otherwise, the like button will be presented, providing the user an option to like an answer. Any answer that has already been liked will be presented with an unlike option instead.

When each answer for a corresponding question is printed, the page will also check if the user is currently logged in to the website. If the user is logged in, then a second query is made to the "likes" table in order to obtain the list of answers that this particular user has liked.

**select** * **from** users, likes
**where** likes.uid = users.uid
**and** users.uid = $uid;

For every answer posted, it will check the answer id and see if this user has liked the post before. If not, then the user will be presented with an option to like the answer which will insert a new value into the "likes" table to show that the user has liked this answer. The web page will send the answer id, the id of the user logged in, and the id of the user who posted the answer into the insert query:

**INSERT INTO** likes(aid, uid, points, r_uid) **VALUES** ($aid, $_SESSION[uid], 1, $r_uid)

Otherwise, the user will be presented with a grayed out button,presenting the option to unlink a post. In addition, whenever a user does like a post and after the web page inserts a new value into the MySQL server, the MySQL server will trigger the following trigger:

**CREATE DEFINER=`root`@`localhost` TRIGGER `project`.`likes_AFTER_INSERT`**
**AFTER INSERT ON `likes` FOR EACH ROW**
**BEGIN**

```
DECLARE temp INT DEFAULT 0;
set temp = (select sum(points)
                    from likes
                    where likes.r_uid = new.r_uid
                    group by likes.r_uid);

if(temp > 20)
then UPDATE users
    SET users.status = "Intermediate"
    WHERE users.uid = new.r_uid;
END IF;

IF(temp > 50)
THEN UPDATE users
    SET users.status = "Expert"
     WHERE users.uid = new.r_uid;
END IF;

update post_answers
set grade = grade + 1
where aid = new.aid;
END
```

Essentially, upon any insert of a new like value, the server will count the total number of likes that the user who posted the answer has received. If it goes past a total of 20 likes, the user's status is updated into intermediate. If the user's total count goes past 50, then that user's status is elevated into expert. Furthermore, the answer that was liked will have its grade, or like counter, incremented by 1. This is because a user's grade attribute is separate from the likes table because of the possibility of other factors which could influence the user's answer's grade.

One such possibility is that if a user's answer is selected as best, then the user's answer's grade will increase, but the number of likes given to that particular user does not change. This means that only likes will have a chance to increase a user's status, but an answer that is selected as best answer will not affect the user's possibility of increasing his/her status, but it will increase the grade of the answer.

However, if a user chooses to unlike a post, then the web page will send a query to delete the likes in the like table:

**DELETE FROM** likes **WHERE** uid = $_SESSION[uid] **and** aid = $_POST[unlike] **and** r_uid = $_POST[r_uid]";

This query will allow the user to remove his/her like from the database which will trigger a delete trigger similar to the trigger posted above except for the opposite purpose. The trigger will decrease the answer's rating to reflect the updated set, and the trigger will count the total number of likes the answer's poster has acquired to see if the user has dropped below the threshold needed for a user promotion. If the user has dropped to one of specified thresholds, the user will be demoted to "BEGINNER" or "INTERMEDIATE" depending on the threshold.

The web page will return back to the answers page where the unliked answer will revert, allowing the user to re-like the answer if desired.

Regarding the best answer functionality, only the poster of the question will be allowed to choose an answer as the best answer. All answers are automatically marked as not the best, and all questions are initially written as unresolved. When the user chooses the best answer, the page will send an update query using the answer id and set the specified answer as the best answer along with increasing the answer's weight and likes score as a reward for having the best answer. This increase in likes will not be considered when a user's status might change. The answer will also be moved to the top of the page and marked in red to indicate the chosen answer.

In addition, the question posted will now be updated to be set as resolved. This will be indicated by the question row being set to bright pink rather than the standard table color. The only way a question can be set to resolved is if a best answer is chosen. Furthermore, there can be only one best answer where if the user chooses another answer to be the best answer when one best answer has already been chosen, then the previously marked best answer will be unmarked and lose its bonus reward. The question will remain resolved, and the user who asked the question is free to answer his/her own question and mark it as the best answer as well.

## LOGIN SESSION

Username login and logout functionality was implemented using PHP's session variables. Whenever a user looks at the site for the first time, the session will not be able to find the user's session username, meaning the user must not be logged in at the moment. When the user enters the login page, he/she will be greeted with a login page. The user will enter his/her login details where upon submitting, the webpage will check the MySQL users table to see if there is a match for username and password.

```
select uid, username, `password`
from users
where username = '$name'
and `password` = '$password'
```

If there is a match, then the PHP session will record the username found and the user will be recognized as logged in as that particular user. Shortly after, the page will forcibly redirect the user to the homepage.  A successful login is shown by the header page where "User Page" will be replaced with the username of the user who logged in. Otherwise, the user will be displayed with an error message saying that the login has failed, possibly due to a mismatch of either the username or password.  Once logged in, gains access to the rest of the site's features such as being able to post questions and answers.  When answers are shown, the table will also include a column giving the user the option to like an answer if the user did not post that same answer.  If the user created the question, the user will also be presented with an additional column to select an answer as best answer.

## Login Page

**Good day!** You are not logged in.

Username

Password

Login  Register

**Registration Page**

From the login page, users can access the registration page by pressing "Register." Users must fill out their username, password, and location to register. The first thing that gets checked when registering is if the passwords match. If they do not match, the account is not made and an error message is displayed. If the passwords match but the username already exists, an error message saying this user already exists shows up.

**INSERT INTO** users(uid, username, password, profile, status, city, state, country) **VALUES** ($uid, '$name', '$password', 'PLACEHOLDER PROFILE', 'BASIC', '$city', '$state', '$country')";

A successful registration will log the user in. An SQL query will insert this new user into the users table along with a placeholder profile and as a BASIC user.

## Create an Account

| | |
|---|---|
| Create Username | |
| Create Password | |
| Retype Password | |
| City | |
| State | |
| Country | |

Register

**Profile Page**

If the user is logged in, then he/she will be able to access his/her profile page. The profile page will provide a table containing the contents of the user's profile including the username, password (length shown by '*'), profile details, status, and location. The textbox will be filled out with the user's profile details as the web page will make a query to the MySQL site using the user's stored session ID. If the user wishes to change the contents of his/her profile, the user can change the textbox for each corresponding detail and replace it with his/her own information.

The user will press save where the web page will make a query to the server to update the users table based on the user's session id.

If the user wishes to change passwords, it directs them to the Change Password Page.

## Profile Page

| | |
|---|---|
| Username | Bobby2 |
| Password | ********* |
| Profile | PLACEHOLDER PROFILE |
| Status | BASIC |
| City | New York City |
| State | NY |
| Country | US |

Save
Change Password

## Change Password Page

Users will be able to change their password by inputting their current password and their new password. The current password must be correct and the new passwords must match for the query to update the user's new password.



## Change Password

| | |
|---|---|
| Current Password | |
| New Password | |
| Re-enter Password | |

Change Password    Cancel

## Browse Page

The browse page will come with a search query along with a list of checkboxes for topics and subtopics. The user can click on one or more topics and subtopics and type any amount of keywords into the search bar. The search bar will take the user's input, create a list of keywords and divide them based off of the spaces in the search bar.

The topics and subtopics are a series of checkboxes in the same POST container with the search bar. Once submitted, any checked topic and/or subtopic will be sent in a POST request with a prefix tag to identify what type of topic in the topic hierarchy the inputted value belongs to. For example:

- Subtopic: Microeconomics->sub_Microeconomics to identify that this is a subtopic.
- Topic: Computer Science->top Computer Science to identify that this is a main topic.

The search function will strip off the prefix and put each topic and/or subtopic into a topic array and/or subtopic array respectively. Afterwards, it will build a search query based on the keywords, the topic list, and subtopic list where it will search for any questions that contains one of the keywords and belongs to one of the selected topics that were specified.

- Help 3 is converted into [Help, 3]
- This means that the search bar will search for any questions that contains the words "Help" or "3."

The page will send the following MySQL query to the server:

"**select** *
**from** questions, post_question, users, subtopic
**where** questions.qid = post_question.qid
**and** post_question.uid = users.uid
**and** questions.stid = subtopic.stid
$searchwords

The page will then provide a list of all questions or that have atleast one of the matching keywords found.

For example, if a user opted to search for any questions in "Botany" or "Computer Science," then the browsing page will create a query to pull all questions that belong to the subtopic of Botany or any question that belong to the main topic of Computer Science.

The answers search query functions the same as the questions search query for the answers table. For this presented answers table, there is no ordering for the selection based on grade, rating, or time posted as the search function did not utilize relevance scoring for its search.

The user may also specify if he/she would want to search for answers, questions, or both. These settings are checkboxes which will send a flag as part of the POST request in which the search page will check which flag has been set to make a query in questions, answers, or both. If neither has been selected, the user will simply be presented with an error message saying that the page wasn't ordered to search for anything.

## User Submissions

When looking at the recently submitted page, the user will be presented with the list of questions and/or answers, if the user has posted any. This table will also present an additional option to edit a question or answer. If the user wishes to edit an answer or question, the user will be presented with a form using the contents of the question or answer.

The questions you have posted

| Title: | Body: | Date: | View Answers: | Edit Question? |
|--------|-------|-------|---------------|----------------|
| HOW TO HACK | Hackerman keeps demanding ransom in bittycoin. How do I pay.... | 2011-11-30 18:56:23 | View More | Edit |
| Difference between loops? | What's the difference between a for loop, a while loop, and a do-while loop? | 2001-06-13 11:44:51 | View More | Edit |

The answers you have posted

| Question Title: | Question Body: | Question's Date Posted: | Your answer: | Liked received: | Time posted: | Edit Answer? |
|-----------------|----------------|-------------------------|--------------|-----------------|--------------|--------------|
| HOW TO GROW PLANT | My crops keep dying when I feed it coca-cola. What am I doing wrong. | 2023-12-09 22:59:59 | Plants cannot subsist off cola products. Try something that isn't lethal. testing | 103 | 2023-12-09 22:59:59 | Edit |

## Edit Question Page

| | |
|---|---|
| Question Title | HOW TO HACK |
| Question Body | Hackerman keeps demanding ransom in bittycoin. How do I pay. |
| Select a topic: | CyberSecurity |

Save

For the edit questions page, an SQL query is made to obtain the entire list of subtopics and identify which subtopic is the user's subtopic. The question's original subtopic will be displayed as the default option while the rest of the subtopics will be write into the dropdown menu.

## Edit Answer Page

| | |
|---|---|
| Answer Body | Try jumping up and down. |

Save

Similar to the profile editing page, the form will be automatically written with information corresponding to the question or answer being targeted. The user may choose to edit one of the forms which will cause the web page to write an SQL update query to the table based on what changes were found.

If no changes were found, the user will be informed that there was nothing to change. Otherwise, the user will be redirected to the recently submitted page to be shown the list of questions and answers, including the newly edited one in question.