

Design and Implementation of a Web Forum

Angelina C. Rotundo

Abstract

Forum as termed in the American Heritage Dictionary is a “public place or medium for open discussion [1].” Web forums are very important in today’s society and are becoming increasingly prevalent in businesses as a common tool for communication. As the internet began to grow and advance significantly in the early 1990’s so did the use of Web 2.0 standards. The standard “forum” of physically meeting in a common location is being replaced by the web forum. This project proposal is to develop a web forum for the Society of Computational Science (SCS) website. The forum will allow users to start and participate in online communication by creating topics of discussion called threads and replying to messages within each thread topic. This paper discusses the design and implementation of a web forum using XHTML, PHP and MySQL.

Index Terms - Apache Server, Attributes, Cascading Style Sheets (CSS), Database, Database Design, Entity, Entity Set, Entity-Relationship (E-R) Diagram, Forum, HTML, Internet Forum, Message Board, MySQL, PHP, Posts, Primary Key, Relational Database Management System (RDMS), Relationship, Structured Query Language (SQL), Thread, Web Forum.



1. Introduction

A *Web Forum*, also referred to as an *internet forum*, is a type of online discussion site similar to a message board [2]. It is usually placed on a website where users of the site can share information. Information on a web forum is organized in a hierarchal structure by topic or subject. Users post messages under topics called threads. The threads can be created by either the administrator of the site or users accessing the site. Messages can be posted as either replies to existing threads or messages or posted as new threads. The site administrator monitors the site to ensure that the thread topics are relevant to the forum. Some websites may have multiple forums. Each forum therefore can have multiple threads posted by multiple users. Web forums can therefore range from very simple to quite complex depending on how many forums and features are included.

The design and implementation of a web forum consists of three key components which are 1) the forum design, which consists of the overall forum design layout and functionality; 2) the forum database, which stores the messages and user information; and 3) the web application pages, which consist of HTML and PHP code.

This paper provides a detailed overview of designing a web forum with an emphasis on the forum structure and tools needed such as PHP and MySQL.

2. Forum Design & Implementation Process Overview

Designing and implementing a web forum requires a thorough understanding of the three components mentioned earlier. The first component, the forum design, is the first step to the process because it involves determining what the forum will look like, what its purpose is, and how it will behave or interact with the users. The forum design therefore consists of the overall forum layout and functionality. The second component is the forum database, which stores all of the information such as messages, threads, user names, etc. Designing the database involves numerous steps which we will discuss in section 3. Having a good database design is critical to the forum because it is where the forum goes to retrieve and store messages. Without the database, the forum could not exist. The third component is the web application pages, which consist of HTML and a programming language called PHP that communicates

with the database to store and retrieve the messages and other information.

The paper walks through the process of designing and implementing a web forum focusing on the following steps:

- Forum Design
 - Purpose
 - Functionality – user interactivity, layout, and features
 - Styles – colors, look and feel, etc.

- Database
 - Tools Needed / Set Up (Apache Server, PHP, and MySQL)
 - Forum Structure (Hierarchical Structure of Messages)
 - Database Design – Storing and retrieving messages in the database
 - Database Implementation - SQL script with attributes

- Web Application Pages
 - PHP pages
 - HTML pages
 - Forum Structure (Hierarchical Structure of Messages)
 - Database Design – Storing and retrieving messages in the database
 - Database Implementation - SQL script with attributes

2.1 Forum Design (Purpose)

The purpose of the Forums is to allow users of the Society of Computational Science to communicate and share information on subjects relevant to the site such as Academics, Algorithms, Imager and Events.

By using the forum, users can do more than just download and read documents and information posted on the site. Computational Science is a broad category so having a forum will give users the ability to communicate about different topics within the computational science subject. SCS Forums will give users the additional advantage of communicating directly with other users of the site. It also allows the Society of Computational Science to receive feedback and input from users which can be used to customize the site to the user needs and preferences.

2.2 Forum Design (Functionality)

SCS Forum was designed with ease of use and functionality in mind. The goal was to design the forum in a way where users could navigate easily through the forum and post messages intuitively. SCS forums can be accessed by clicking the “Forums” link on the main navigation menu of the SCS site. The SCS Forums homepage is shown below in Figure 1. Most of the information the user needs can be accessed from this main forums page.



Figure 1. Forum Home

The SCS Forums layout consists of a left navigation menu which contains the main links for the forum including links for creating forums, selecting forums, and viewing forum threads and messages. When the “View Forum Threads” link is selected, the available threads within the selected forum are displayed on the right as shown in Figure 2 below. Each thread has a subject, email of the person who posted it, date the thread was posted, date of the last post, and number of replies made to the thread.

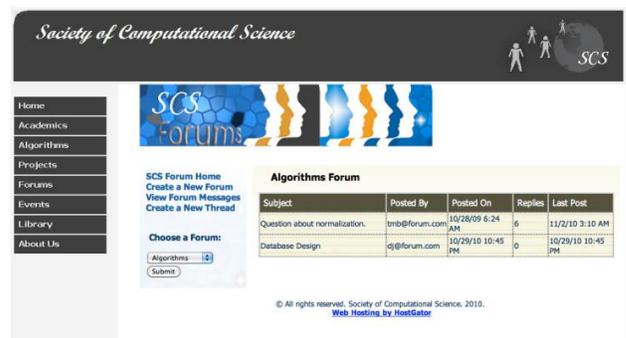


Figure 2. View Forum Threads

As mentioned earlier, some web forums consist of only one forum. SCS Forums was designed with multiple forums in mind so users can choose from a selection of different forums rather than being limited to only one. The available forums can be viewed by using the drop-down list under “Choose a Forum” as shown in Figure 3. If a specific forum topic is not in the list of forums then the user can create a new forum specifically for that topic.

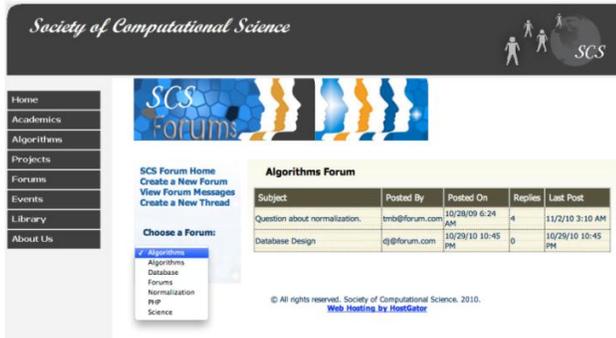


Figure 3. Choose a Forum Drop-down List

If a user wants to create their own forum rather than choosing from a pre-existing list, they can do so by selecting the “Create a New Forum” link. Figure 4 below shows the screen for creating a new forum.



Figure 4. Create New Forum

The forum name and user information will have to be provided. Once the new forum is created it will appear in the drop-down list as shown in Figure 5. In this example, we created a new forum called “New Forum.” As you can see in Figure 5, “New Forum” is in the list of available forums.



Figure 5. View New Forum in Drop-down List

When a new forum is created, the list of threads on the right will be empty so a new thread will have to be created as shown in Figure 6 below. A new thread can be created by selecting the “Create a New Thread” link. It should be pointed out that this link for creating a new thread is not listed on the SCS Forums home page and is only available when the user is on the page for viewing forum threads.



Figure 6. Empty Thread List

Once the “Create a New Thread” link is selected, a user can create a new thread by filling in the form as shown in Figure 7. In this example, we created a new thread called, “New Thread for the New Forum.” After the form is filled out and the information is processed, by clicking “View Forum Messages” link, we can see the new thread we created as shown in Figure 8.



Figure 7. Create a New Thread

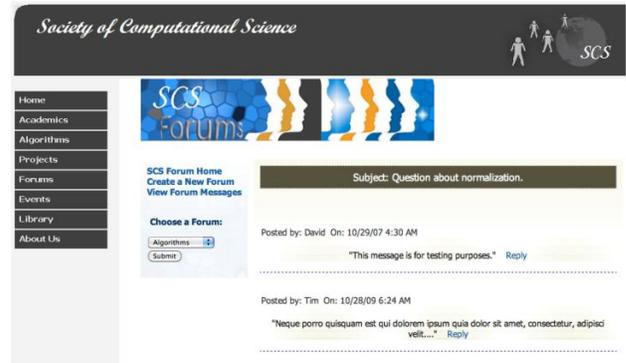


Figure 9. View Forum Thread Details



Figure 8. View New Forum



Figure 10. Add a Comment to Selected Thread

As we discussed earlier, the list of available threads within a forum as shown in Figure 2 is the available subjects or conversations within the forum. By clicking a thread subject, the list of messages within that particular thread will be displayed as shown in Figure 9 below. At the top of the page, the thread subject is listed. In this example, the thread is called, “Question about Normalization.” For each message within a thread, SCS Forums will display the message body, the first name of the person who posted it, and the date and time it was posted. At the bottom of this page there is a form for users to add a comment to the thread subject as shown in Figure 10 below.

If a user wants to reply to an individual message within a thread, they can do so by clicking the “Reply” link next to the message. As Figure 11 shows below, the user will be prompted to add their reply and click “Submit.”

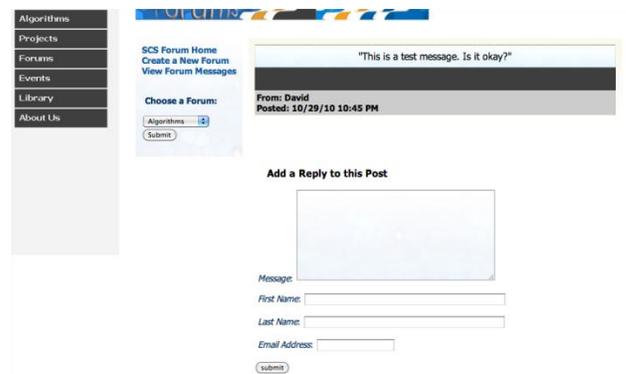


Figure 11. Post a Reply to a Message

After a user submits a reply or creates a new forum or thread, they are presented with a confirmation screen as shown in Figure 12 below which confirms that their information has been processed successfully. If the information was not processed successfully then they would receive an error message instead.



Figure 12. Post a Reply Confirmation Screen

2.3 Forum Design (Styles)

Once the purpose of the forum was determined and the functionality and basic layout decided on, the next step was to work on the forum styles. Adding styles to the SCS Forums consisted of designing the forum look and feel. This process included adding the styles such as colors, font sizes and styles, margin specifications, etc. The colors chosen for SCS Forums complement the colors of the main Society of Computational Science website. Rather than including the styles in every PHP and HTML page, they are maintained in one file called *ForumStyle.css*. The extension *.css* stands for **Cascading Style Sheets (CSS)** and is a quick and efficient way of applying and maintaining styles for a web site. If a style needs to be updated, it only needs to be updated in the main CSS file. This is why it is referred to as “*cascading*” because it cascades down to the other pages.

In addition to creating the styles, a SCS Forums logo was also created, which ties in the original SCS logo to the “forum” concept of “*multiple people communicating across the globe but in one central location.*”

3. DATABASE

3.1 Tools

The tools used to develop SCS Forums include PHP, MySQL, and Apache. **PHP**, which stands for *PHP Hypertext Preprocessor*, is a server-side scripting language that is used for web development [4]. The PHP

code is embedded into the HTML pages and executes whenever a user visits the HTML page. The “PHP code is interpreted at the web server and generates HTML or other output that the visitor will see” [3]. The web server application used for this project is Apache. PHP is open source and has many advantages including the following [3]:

- Very fast
- Performance
- Scalability
- Interfaces to many database systems
- Built in libraries for common web tasks
- Easy to learn
- Strong object-oriented support
- Low cost

“In order for the PHP scripts to retrieve information from a database, it must have a database [3].” Databases are used to store, search, sort, and retrieve data [3]. A little later we will walk through the development of the database for SCS Forums. MySQL is a **relational database management system (RDMS)**. It is the most popular open-source database in the world [4]. A relational database uses tables to store information. “Each table has multiple columns and each column has a unique name [6].” MySQL uses **Structured Query Language (SQL)**, which is the standard database query language, to store and retrieve data from the database. By using PHP and MySQL we can quickly store and retrieve information submitted and requested on the SCS Forums web site pages.

3.2 Installation

You can install Apache, MySQL, and PHP individually or together. For this project, MAMP was used to do an all-in-one installation of Apache, MySQL, and PHP on a Mac OS X. MAMP is free software that stands for **Mac OS X**, the operating system, **A**pache, the Web server, and **M**ySQL, the database management system or database server, and **P** for PHP used for the web development [2].

3.3 Forum Structure

The structure of a forum usually depends on the specific software requirements and user needs. There are many different types available, where some are very complex and others are simple in nature. One of the first questions to ask when designing a web forum is if there is a need to have more than one forum. For example, suppose the information shared on the forum ranges across many

different topics. In this case, grouping this information into categories or separate forum topics is probably going to be more beneficial. By creating separate forum topics, we can provide the users with faster access time and increase their ease of use in finding information.

The structure of messages in the SCS Forum can be described as a hierarchy structure with Forums at the top of the hierarchy. The forums are simply separate topics which are like containers that organize the messages. As mentioned earlier, SCS Forums allows users to select a forum from a drop down menu. In this way, we can have multiple forums each with completely different topics. Rather than having different topics all mixed together, it is much more efficient to store the different topics in separate Forums. For example, SCS Forums begins with some test Forums called Algorithms, Database, Forums, Normalization, PHP, and Science. Within each Forum, users can post messages, which begin as Threads. Threads are the next level down from Forum and are subjects or topics. More simply put, each forum is further organized into separate threads or topics. The next level down from Threads is Posts, which are the messages. Each thread can have multiple posts, which are comments or messages to each thread. The last level is Children, which are posts that are also replies. The logic is fairly straightforward as described in Figure 13.

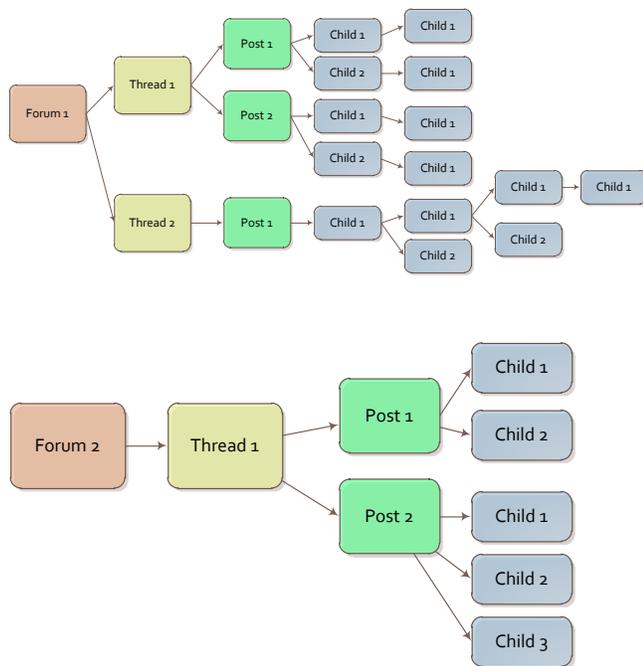


Figure 13. Structure of Messages in SCS Forum

3.4 Database Design

Designing the forum database begins with identifying the tables and attributes and how they work together to store and retrieve messages. When users post replies and create threads, the data is stored into a database. When users view the forum page, messages are retrieved from the database and displayed onto the screen.

3.4.1 Database Overview

The SCS Forums database consists of 4 tables: 1) *forums*; 2) *threads*; 3) *posts*; and 4) *users*. The forum topics will be stored in the *forums* table. The thread information which includes the title and message body will be stored into two tables. The *threads* table stores the thread title or subject. The *posts* table stores the actual message body. Storing the thread information into two tables in this way provides the benefit of locating thread data easily and not bogging down the *threads* table with message text. Furthermore, we want the users to provide their name and email address when posting topics so a table called *users* was created to store the user information.

One of the nice features of SCS Forums is that it allows users to select and create forums. In order to allow users to select and create forums, a table called *forums* had to be created, which stores the forum information. The only forum information we want to store is the forum name so we have an attribute called *forum_name*. When a user creates a new forum, the forum name is stored in Table *forums* and the forum is automatically assigned a new *forum_id*. Each time a new forum is created the *forum_id* automatically increments. In this way, the *forum_id* is a unique attribute referred to as the **primary key** of the database table. The SCS Forums Administrator can create pre-existing forums, which the user can select via a drop-down list on the SCS Forums home page as shown in Figure 3.

Each message is a post so all messages are stored in the table called *posts*. When a message is posted to the *posts* table, it is automatically assigned a *post_id*, which is also the primary key for the *posts* table.

As mentioned earlier, the message hierarchy directly under the Forum level begins with Threads, which are initial posts to the forum. Threads are subjects within a forum. Since only threads have subjects, rather than assigning the threads to the *posts* table, it is more practical and efficient to store them in their own table called *threads*, which will store only the subject of the thread and the user email. When a new thread is created, information is posted to three tables. The subject is

inserted into table *threads*, the user information is posted to the *users* table, and the body of the message is posted to table *posts*. Each thread is a post, but the body of the message for a thread is stored in the *posts* table, which allows for faster access time when searching through the *posts* table. The *threads* and *posts* tables are linked together by the *thread_id*, which is the primary key for the *threads* table.

The table called *users* contains user information. It stores a user’s first name, last name, and email. The *email* is the primary key of table *users*, which relates to *table threads* and *table posts*. Each time a user creates a post, their first name, last name, and email information is stored in the *users* table. SCS Forums does not require users to register or login.

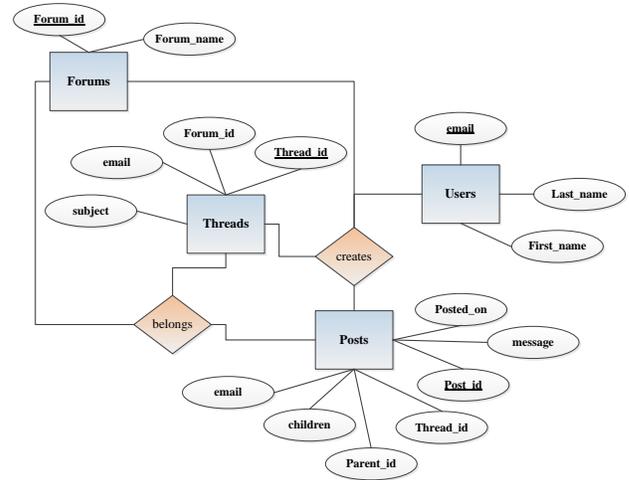


Figure 14. Entity-Relationship (E-R) Diagram

3.4.2 Entity-Relationship Diagram

Database design involves the design of the database schema, which is the overall design of the database [6]. “An **entity** is a “thing” or “object” in the real world that is distinguishable from all other objects [6].” For example, each user in SCS Forums is an entity. “An **entity set** is a set of entities of the same type that share the same properties, or attributes [6].” In our example, one entity set is *users*, which represents the set of all users in the SCS Forums. Another entity set is *forums*, which represents the set of all *forums* within SCS Forums. “**Attributes** are descriptive properties possessed by each member of an entity set [6].” Each entity in an entity set has a value for each attribute [6]. For example, the attributes for the *forums* entity set are *forum_id* and *forum_name*. The attributes for the *users* entity set are *email*, *last_name*, and *first_name*.

An **entity-relationship (E-R) diagram** is used to graphically represent the database schema. The entity-relationship (E-R) diagram for the SCS Forums database is shown below in Figure 14. As the E-R diagram shows, the database tables or entities each have primary keys, which are underlined. As mentioned earlier, primary keys are unique attributes of an entity set. The tables are linked to each other by these primary keys and by relationships. A **relationship** is an association among several entities [6]. In this example, we use two relationships: 1) *creates*; and 2) *belongs*. The users create forums, threads, and posts. Each post belongs to a thread, which belongs to a forum.

3.4.3 Message Storage Structure

Table 1 below shows the attributes of the *posts* table and how a message can have a parent and children. When a message is created it is assigned a *post_id*, which is a chronological auto incrementing integer. If the post is an initial post, then its *parent_id* will be zero since it doesn’t have a parent and the *children* attribute will also be zero since it’s not a reply. However, if the post is a reply to another post, then its *parent_id* will be equal to the *post_id* of the post it’s replying to and the *children* attribute will increment by one. Table 1 provides a few examples. In this example, Post 1 has two *children* but a *parent_id* of zero because it is the initial post. Post 5 has a *parent_id* of 4 because it is a reply to Post 4. Post 5 has zero *children* because no user replied to Post 5. It should also be noted that every post is assigned a *thread_id* because posts in the hierarchy structure are one level below the thread level. Even if a post is a new thread, it will still have a *thread_id*. Every thread is in turn linked to a specific forum.

Table 1. Message Structure

post_id	thread_id	parent_id	children	email	message
1	1	0	2	tc@test.com	Post 1's initial message
2	1	0	1	markb@test.com	Post 2's initial message
3	1	1	1	paul@site.com	Post 3's initial message, which is a Reply to Post 1
4	1	0	2	jav@email.com	Post 4's initial message
5	1	4	0	123tom@team.com	Post 5's initial message, which is a Reply to Post 4
6	2	1	0	trz@you.com	Post 6's initial message, which is a Reply to Post 1
7	3	4	0	tc@test.com	Post 7's initial message, which is a Reply to Post 4
8	3	2	0	markb@test.com	Post 8's initial message
9	4	3	0	jav@email.com	Post 9's initial message, which is a Reply to Post 3

3.5 Database Implementation

After the database design was decided on and the entity-relationship diagram was created, the SQL script to create the database and generate the tables was created. As mentioned earlier, SQL stands for structured query language. In addition to querying a database, SQL can define the structure of the data, modify the data in the database, and specify security constraints [6]. The specific tool used to generate and manage the database is called MySQL Workbench. MySQL Workbench is a unified tool used to design, model, generate, and manage databases. There are other tools on the market, but this one fits our needs primarily because it is free and easy to use.

The SQL script for defining the four database tables is shown below in Figure 15. In this database all of the primary keys auto increment except for the *users* table primary key, which is an email address.

The SQL script also contains test data, which was used to populate the tables so the SCS Forums could be tested. The test data was inserted into the tables using the MySQL INSERT INTO script. For example, we created some base forums and user information so the SQL script for populating table *forums* and table *users* is shown below in Figure 16.

```
CREATE TABLE users (  
  first_name VARCHAR(20) NOT NULL,  
  last_name VARCHAR(40) NOT NULL,  
  email VARCHAR(80) NOT NULL,  
  PRIMARY KEY (email)  
);
```

```
CREATE TABLE threads (  
  thread_id INT UNSIGNED NOT NULL  
  AUTO_INCREMENT,  
  forum_id TINYINT UNSIGNED NOT  
  NULL,  
  email VARCHAR(80) NOT NULL,  
  subject VARCHAR(150) NOT NULL,  
  PRIMARY KEY (thread_id),  
  INDEX (forum_id),  
  INDEX (email)  
);
```

```
CREATE TABLE posts (  
  post_id INT UNSIGNED NOT NULL  
  AUTO_INCREMENT,  
  thread_id INT UNSIGNED NOT NULL,  
  parent_id INT UNSIGNED NOT NULL,  
  children int default 0 NOT NULL,  
  email VARCHAR(80) NOT NULL,  
  message TEXT NOT NULL,  
  posted_on DATETIME NOT NULL,  
  PRIMARY KEY (post_id),  
  INDEX(parent_id),  
  INDEX(children),  
  INDEX (thread_id),  
  INDEX (email)  
);
```

Figure 15. SQL Script for Creating forumSCS Tables

```
INSERT INTO forums (forum_name)  
VALUES  
( 'Algorithms'), ('PHP'), ('Forums'),  
( 'Database'), ('Science'), ('Normalization');  
  
INSERT INTO users (first_name, last_name,  
email) VALUES  
( 'Tom', 'Brown', 'tmb@forum.com'),  
( 'David', 'Jones', 'dj@forum.com'),  
( 'Gary', 'Brenan', 'gk@forum.com'),  
( 'Tim', 'Smith', 'tc@forum.com'),  
( 'Beth', 'Hayes', 'bh@forum.com');
```

Figure 16. MySQL INSERT INTO for defining test data

Once the SQL script was written it was executed to generate the database and tables. The tables and corresponding attributes are shown below in Figure 17.

```
CREATE TABLE forums (  
  forum_id TINYINT UNSIGNED NOT NULL  
  AUTO_INCREMENT,  
  forum_name VARCHAR(60) NOT NULL,  
  PRIMARY KEY (forum_id),  
  UNIQUE(forum_name)  
);
```

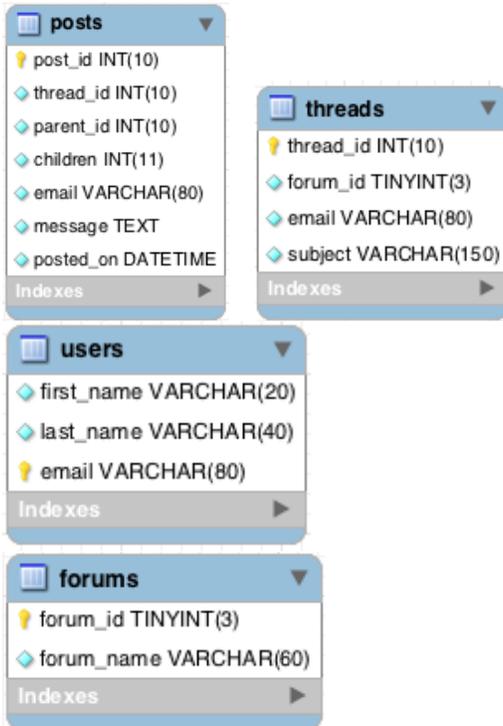


Figure 17. SCS Forums Database Tables

4. Web Application Pages (PHP)

Having a forum design and database is only one part of creating a web forum. In addition to the layout and database design, the forum requires application pages which are comprised of the HTML and PHP code.

4.1 PHP

With PHP installed, your web server will be able to run PHP scripts that can do tasks such as retrieving up-to-the minute information from a database and using it to generate a web page on the fly before sending it to the browser that requested it [3]. The PHP pages used in SCS Forums allow users to view, post, and reply to messages. It also allows them to create new forums. SCS forums consist of the following PHP pages. Each page has its own specific function.

- forum.php – Displays the threads in the forum selected; access it by clicking “View Forum Messages”
- index.php – SCS Forums home page
- mysqli_connect.php – Establishes a connection to the database

- post.php – Processes the message post and displays the form for creating a new thread
- post_form.php – Displays the form for posting threads
- post_form_reply.php – Displays the form for replying to posts
- post_reply.php – Processes post_form_reply.php
- post_Forum_form.php – Displays the form for creating new forum
- post_Forum.php – Processes post_Forum_form.php
- read.php – Displays the threads and posts in a forum
- read_reply.php – Displays the replies in a forum
- val_functions.php – Contains data validation functions
- view_functions.php – Contains functions for displaying the data
- include_functions.php – Includes the functions

One of the most important PHP pages listed above is the `mysqli_connect.php` page because it establishes the database connection. It does this by connecting to the MySQL server with PHP using a built-in function called `mysqli_connect`. The `mysqli_connect` function is written as follows:

```
mysqli_connect (hostname, username, password)
```

The other important part of the PHP pages is the SQL queries that are used within the PHP code to retrieve the data and display the results or store the data to the database when it is submitted. For example, the `read.php` page uses a SQL query to display the threads in a forum. The `post.php` page processes the data when a message is submitted. This page uses a SQL `INSERT INTO` statement to insert the data into the database tables. The PHP code uses a “POST” message to receive the information when it is submitted and a “GET” message to retrieve the information when it is requested. The PHP code in conjunction with the Apache Server stores and retrieves the information in the MySQL database tables. This allows users to submit information and view information within the forums.

4.2 HTML

In addition to the PHP pages listed above, the main site web pages were also included as special include files within the PHP pages themselves. By including the HTML pages as header and footer includes, we avoid the need to retype the HTML code for the entire site on the forum pages. The header include includes all site

information except for the footer and is placed at the top of the PHP page. The footer works the same way as the header, but contains the footer HTML information and displays the web site footer. The include statements for the header and footer HTML files are written as follows:

```
include ('includes/footer.html');  
include ('includes/header.html');
```

5. Results

SCS Forums was implemented using HTML (localhost) web page, Apache Server, PHP, and MySQL. The Forums pages were tested thoroughly to ensure that the PHP code was working by getting user input, storing it in the database, and displaying it on the screen. Ensuring that the database connection was established was very important, otherwise, the forum PHP pages would not be able to POST and GET data from the database. The database design process required some careful planning to ensure that messages were getting stored into the database tables and the correct message hierarchy was being used as shown in Table 1.

6. Conclusion

Web forums are very instrumental tools used to share information. By setting up a web forum, users from across the world can share information in one central location. SCS Forums was designed to allow the users of the Society of Computational Science share information. Because SCS Forums has the advantage of using multiple forums, the administrator of SCS Forums as well as the users can create forums according to their needs. By organizing SCS Forums into separate forums users can easily find the information they are looking since the information is organized by subject. Allowing users to communicate in a web forum is an excellent and valuable addition to any web site, business or personal.

References

- [1] *The American Heritage Dictionary*. 4th ed. New York: New York: Dell Publishing, p 221, 2001.
- [2] “Internet Forum,” Wikipedia, 2010. [Online] Available: http://en.wikipedia.org/wiki/Internet_forum. [Accessed: November 1, 2010]
- [3] Luke Welling, Laura Thompson, *PHP and MySQL Web Development*. 4th ed. New York: New York: Pearson Education, Inc., p. 3. 2009.
- [4] Larry Ullman, *PHP 6 and MySQL for Dynamic Web Sites*. Berkeley: California: Peachpit Press, 2008.
- [5] “MAMP,” Wikipedia, 2010. [Online] Available: <http://en.wikipedia.org/wiki/MAMP>. [Accessed: November 1, 2010]
- [6] A. Silberschatz, H. Korth, S. Sudarshan, *Database System Concepts*. 6th ed. New York: New York: McGraw-Hill Companies, Inc., pp. 3, 16. 2011.
- [7] Kevin Yank, *Build Your Own Database Driven Web Site Using PHP & MYSQL*, 4th ed. Collingwood, Vic: Australia: SitePoint.
- [8] “PHP,” The PHP Group, 2001. [Online] Available: <http://www.php.net>. [Accessed: November 1, 2010]
- [9] “Database Design, Implementation, and Normalization,” Society of Computational Science, 2010. [Online] Available: <http://www.societycomputationalscience.org>. [Accessed: July 22, 2010]

Biography

Angelina C. Rotundo is a graduate student at Florida Institute of Technology, Orlando Graduate Center. She works as an engineer for the Space Shuttle Program located at the Kennedy Space Center, Florida. Her research interest is in web development and database design.