

MSG - Online Message Forum

Gordon Swan

40202556@live.napier.ac.uk

Edinburgh Napier University - Advanced Web Technologies (SET09103)

Abstract

One of the most important uses for the internet has been communication, from the early days of using Internet Relay Chat for text-based messaging to modern social media giants such as Facebook and YouTube, sharing high definition multimedia content with large numbers of people at once, the method of sharing and communicating information with others has always been a keen interest when developing for the web. One such method of sharing information is the forum, a virtual public notice board, mostly suited to question and answer style situations, such as tech support. Internet forums are extremely popular even today, with Stack Exchange[1] being one of the largest internet forum networks in existence.

Keywords – forum, board, post, thread, text, image

1 Design

Database Design In order for MSG to store any information long term, it requires access to a database. Python has an SQLite Module, which can be used to read and write database files very efficiently. This method of data storage is far superior than using JSON or text file based storage solutions as SQLite is far more efficient and much faster. The database itself consist of three tables, one for the threads, one for the comments on the threads, and one for the user accounts. The tables are also linked together using foreign keys, this helps to associate posts with threads, and then with the owning user accounts. This is especially useful if content needs to be erased from the database as all linked content can also be removed to reduce fragmentation. The Database Schema can be seen in Listing 1.

Listing 1: SQLite Database Schema

```

1 CREATE TABLE IF NOT EXISTS thread (
2   thread_id integer PRIMARY KEY AUTOINCREMENT,
3   thread_title text,
4   thread_content text,
5   thread_img text,
6   thread_owner varchar,
7   thread_active boolean DEFAULT TRUE,
8   thread_time integer
9 );
10
11 CREATE TABLE IF NOT EXISTS post (
12   post_id integer PRIMARY KEY AUTOINCREMENT,
13   post_thread integer,
14   post_content text,
15   post_img text,
16   post_owner integer,
17   post_active boolean DEFAULT TRUE,
18   post_time integer,
19 
```

```

20 FOREIGN KEY (post_thread) REFERENCES thread (
21   thread_id)
22 FOREIGN KEY (post_owner) REFERENCES user (
23   user_id)
24 );
25
26 CREATE TABLE IF NOT EXISTS user (
27   user_id integer PRIMARY KEY AUTOINCREMENT,
28   user_level integer DEFAULT 0,
29   user_name varchar,
30   user_email varchar UNIQUE,
31   user_password varchar,
32   user_active boolean DEFAULT TRUE
33 );

```

UI Design The User Interface has been designed to conserve space by using modals for the 'New Thread' and 'New Post' forms. This design choice means that more of the screen space can be dedicated to the forum content, instead of data fields that aren't necessarily being used all the time. This is especially important if the user is on a mobile platform, where screen space is often very limited. In addition to saving screen space, modals provide a modern, responsive feel to the site's performance, while still performing well on low power devices. The responsive nature of the site, as well as the modal functionality is all thanks to the Bootstrap Framework[2] providing the majority of the user interface components for the site.

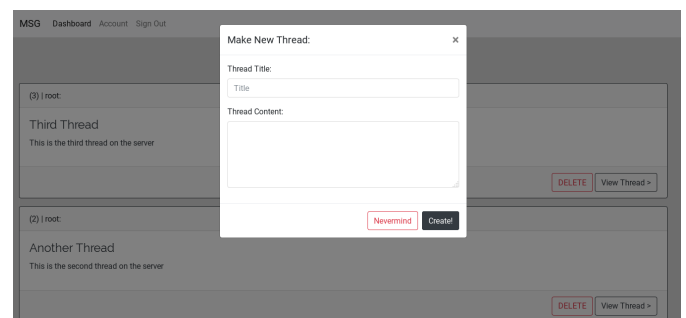


Figure 1: - Modals - Modals used to hold forms

2 Enhancements

Multiple Boards One particular area where MSG is lacking functionality is with multiple board support. Most, if not all modern forum sites allow for multiple boards covering different topics. This is most definitely a missing feature from MSG, however with some tweaking could be implemented. For instance the database schema could be improved to feature an additional table which hosted different board topics,

and boards could be linked to this table via foreign key.

Multimedia Support In some message boards, it is possible for the user to upload multimedia elements, such as images and videos. A possible future feature of MSG could be multimedia support, for hosting short videos, and images for the duration of the thread's lifetime. One considerable challenge with this feature would be maintaining security, as all content being uploaded to a server by a user should be considered malicious until tested and evaluated to be otherwise. Therefore, creating sanitation software that ensures security of the site is maintained would be of some difficulty.

3 Critical Evaluation

Security With so many websites now storing information about their users, it's important for these websites to employ some level of security and encryption to reduce the impact of data breaches and leaks, and to prevent unauthorized access to sensitive data in the first place. MSG makes use of password hashing to keep user passwords secure, even if the database file is compromised by a malicious entity. This is because password hashing compares hashed keys rather than plain text passwords, avoiding the need to ever store or compare actual passwords at all.

In addition to password hashing, MSG also uses tag stripping to remove any HTML or SQL code from user-entered content. This reduces the risk of SQL Injection attacks and Cross-Site scripting as it is much harder for a malicious user to execute code snippets on the site without authorization. Without tag stripping, a user may be able to insert PHP/Jinja2/SQL source code into a field which would be displayed as post or thread content, and have that code executed when the page - now containing malicious code - is rendered by the app. Listing 2 shows the tag stripping method, adapted from Stack Overflow (Another popular internet forum part of Stack Exchange)[3]

Listing 2: Python HTML Strip Tags

```
1 class MLStripper(HTMLParser):
2     def __init__(self):
3         self.reset()
4         self.fed = []
5     def handle_data(self, d):
6         self.fed.append(d)
7     def get_data(self):
8         return ''.join(self.fed)
9
10    def strip_tags(html):
11        s = MLStripper()
12        s.feed(html)
13        return s.get_data()
14
```

Administrative Tools All internet forums, which operate around a public based community, require some level of administration and moderation to combat abuse and misuse. MSG has some features for allowing elevated users to control content such as moderator status, which permits trusted users to delete content that they believe to be inappropriate. These features are however, lacking as there is currently no functioning ban list for keeping undesirable users out of the site. The only step in this direction is user account control,

in which accounts can be deactivated, but there is not yet any restriction on creating new or multiple accounts.

4 Personal Evaluation

Database Initialization One particular challenge has been understanding and learning to use the database schema system, which allows for a database to be initialized from a script before being used with the web application. This is extremely useful as it allows the web application to be shipped or deployed on a different system, without having to also transfer or copy large database files. It also means that the database can be dropped and re-established easily, without needing to manually construct the database via numerous commands when a new database file needs to be created.

Validation Another challenging concept has been ensuring that all user input is validated correctly, to ensure that users cannot exploit the site and perform operations that they should not be allowed to do such as delete content without the correct level of authorization. As mentioned previously, all user input is stripped of tags to ensure the risk of CSS and Injection attacks are reduced as much as possible. In addition to this, session data should be checked at all times to ensure that a user has the correct permissions to create or delete content on the server, otherwise a user may be able to cause grief for other users by hijacking their threads or deleting their posts.

5 Conclusion

To conclude, MSG effectively makes use of the flask framework and it's extended features to provide a robust and secure public forum. With a simple, yet effective database, and security-focused application structure, MSG effectively achieves it's goal of providing an internet forum web application. There are numerous features which, with more time could make this application far more functional, and arguably more useful compared to other internet forums, however as a simple question and answer system it meets the requirements

References

- [1] "Stack exchange communities." Stack exchange is an on-line forum community for asking and answering questions on a range of topics.
- [2] J. T. Mark Otto, "Bootstrap - the most popular html, css, and js library in the world."
- [3] "Html strip tags." Stripping HTML Tags: <https://stackoverflow.com/questions/753052/strip-html-from-strings-in-python>.