

Swigg

An Interactive Drink-Sharing Community

Nirav Sanghani
Brynn Shepherd
Roman Pedan

Project Overview

Swigg.com serves as a central portal for sharing drink recipes among beverage enthusiasts. Registered Swigg users (Patrons) are invited to share their own drink recipes by posting on Swigg, to show their appreciation for other patrons' drinks by voting (Swiggging) for other patron's entries, or to share their enthusiasms or concerns by posting their thoughts as a comment to the drink recipe (Toasting). Unregistered Swigg users (Guests) are also invited to browse the drink recipes that have already been posted but are not permitted the intoxicating powers of Swiggging, Posting, or Toasting. Guests are also restricted from viewing drink recipes that Swigg Patrons have marked as private.

Search

Swigg allows its Patrons to quickly find their favorite drink recipe by implementing a powerful yet efficient search feature.

Inverted Index

In order to facilitate efficient searching, Swigg uses inverted indexing over all of the posted drink recipes. Whenever a Patron posts a new drink recipe, Swigg immediately indexes it. For each word in the title, description, and keywords of the recipe, Swigg adds the tuple (word, storyID, state, location) to the database. In that tuple, word refers to the word that Swigg is indexing, storyID refers to the unique number that identifies each drink recipe, state identifies whether the word is in the title (0), description (1), or the keywords (2), and location identifies where in that part of the drink recipe the word is located. For example, ('hello', 8, 1, 3) means that the word hello is the third word in the description of the drink recipe with ID 8.

Swigg's inverted indexer intelligently avoids commonly used words such as "the", "in", or "of", since these do not add to the content of a drink. It also filters out all punctuation marks so that these do not prevent certain terms from being searchable.

Search Functionality

Swigg's search function ranks drinks in descending order ranked by the number of occurrences of the search terms in the entire drink entry (which includes the drink's title, description and keywords). Therefore a drink whose entry contains the most occurrences of the words in the search terms then it will be the first result.

In order to ensure the integrity of its search results, Swigg crawls each drink recipe link on a weekly basis to ensure that it is the link is still active. If the page that the link points to is no longer accessible then Swigg removes the drink and the link from its database.

By default, Swigg's homepage pre-searches Swigg entries based on popularity, time, and category. For its Patrons' convenience, Swigg displays the three most recently posted and most popular drinks for each category, including the three most popular and recently posted drinks across all categories.

Database Schema

In order to serve its Patrons most effectively, Swigg uses 7 tables in its schema (described below). In each of these tables, we used an integer 'id' attribute as the primary key.

The **drinks_category** table stores the name of each drink category.

The **drinks_drink** table stores the url of each drink, the owner_id (foreign-key) of the patron who posted the drink, whether or not the drink should be private, the title of each drink recipe, a date/time stamp of every drink recipe, a description of the recipe, the number of votes of each drink, the keywords that describe this recipe, the category_id (foreign-key) of the category that this recipe belongs to, and views stores the number of times that the drink has been visited. A new entry to drinks_drink is made anytime a patron adds a new drink recipe to Swigg.

The **drinks_comment** table stores the comments that have been posted for drink recipe with the id, drink_id (foreign key). It includes the poster_id (foreign key) to uniquely identify the patron that posted the connected. *****Finish this.....*****

datetime identifies the time at which the comment was made and text identifies the text of the comment. An entry to drinks_comment is added anytime a patron posts a new comment on Swigg.

The **drinks_vote** table stores information about the votes made on a particular drink recipe. Voter_id (foreign key) identifies the patron who voted, drink_id (foreign-key) identifies the drink that was voted on, and datetime when the vote was made. There is an entry in drinks_vote for every vote made on any drink recipe on Swigg.

The **drink_word_index** table stores the inverted index that aids in searching for articles. Word is the word that is being indexed. Drink_id (foreign key) identifies the drink recipe that the word is in. The state identifies whether the word can be found in the title, description or in the keywords. Location identifies where, within the given state, the word can be found. There is an entry in drink_word_index for every word (including repeats) that was posted with a drink recipe on our site.

The **accounts_user_profiles** table stores the profile information about each Swigg patron. User_id (foreign key) identifies the patron that this profile is describing. Birthday identifies the date of birth of the patron, location identifies the patron's current location and profession identifies the patron's profession. This table is updated every time a new account is created or when any current account is updated.

Extra Credit Features

In order to serve its Patrons best, Swigg implements several additional features that other products may not offer.

Threaded Comments

Swigg implemented threads in its commenting system. This system allows Swigg Patrons to reply to specific comments made by other Swigg users in a drink recipe, instead of restricting them to reply only to the drink recipe itself.

In order to avoid the clutter that can be found in the busy commenting systems of website such as digg.com, Swigg has limited comment threading to one level. This restriction, however, could theoretically be lifted easily, given enough demand from the Swigg community.

Popular Links

Swigg tracks the top 5 most shared websites and provides a list of them on its homepage. Swigg updates this ranking every time a patron Swiggs a drink recipe or when Swigg that a link has become inactive.

Swigg's Finest

In a section of our site titled *Swigg's Finest*, Swigg publishes the usernames of the Patrons that, on average, have contributed the most popular drink recipes to Swigg. This provides Swigg Patrons with an incentive to contribute only the most delectable of concoctions and allows other Swigg Patrons to admire these highlighted Patrons by giving them the opportunity to visit their profile.

Technologies Used

Swigg utilizes Django, Python, and SQLite in order to fuel its services.

Django – This powerful Web-development framework allows for the efficient development of web-based applications. By fusing html, CSS, python, and SQLite into one platform, Django centralizes and streamlines the development process, while simultaneously providing its users with a robust set of features and enhancing the experience with an intuitive user-interface.

Django provided Swigg with three important advantages over other web-development solutions:

- Django is also optimized to allow for clean and elegant URLs, allowing us to avoid the clutter and confusion that many site's URLs suffer from.
- Django's template system allows us to integrate our design and Python code

- Django ships with a lightweight server, allowing us to quickly test Swigg locally on our PCs. The server recognizes and updates to changes as we make them, saving us from restarting the server after each change that we make.

Python – As a high-level, cross-platform, object-oriented programming language, Python provides the ideal language to satisfy Swigg Patron’s needs. Python offers the ability to rapidly develop powerful web-applications.

SQLite – SQLite is a very lightweight database management system. Using SQLite allows Swigg to develop its website without the need to run a server on the development PC.