Implemented solution:

Technologies used:

PHP - for offline script and website's server side code
MySQL - database
Html + JavaScript - website's client side code
jQuery + Ajax - jQuery library and Ajax used to update the webpage dynamically

There are 2 components implemented:

- Offline script that updates database with recent tweets for a given hashtag [TweetFetcher.php]
    - The script is written in PHP
    - It can be invoked as and when required, either from a web browser, or from the curl command
    - A continuous job can be scheduled to run this script via curl, so that the database gets updated periodically
    - The script uses Twitter's public SDK to fetch recent 15 tweets for a given hashtag.
    - Authentication tokens are acquired by appropriate registrations on twitter's dev api website
- Online webpage that refreshes itself with recent tweets for a given hashtag
    - The website asks for a hashtag from the user
    - It then searches recent tweets for the given hashtag from database [TweetAccesor.php]
    - Periodically after a configured interval, the tweets data auto refreshes itself
    - During auto refresh, the same query is executed, using jQuery and Ajax

Database contains 2 tables:

- Tweets table
    - It has tweets for all the hashtags for which the offline script was run
    - It is keyed on a unique tweet-id
- hashtags table
    - It has all the hashtags for which the offline script was run
    - Each hashtag has a comma separated list of recent tweet-ids, so as to query the tweets table