Project : Poodle.com

Student name : Himarsha R Jayanetti

Student UIN : 01160219

1. Overview

List specifications of this milestone and briefly talk about whether each specification is fulfilled or not.

Table 1: Overview of status for Milestone 2 specifications.

Fulfilled	#	Description
Yes	1	Users should be able to get a confirmation email to verify their email addresses
Yes	2	The website has an “Advanced Search” in which users can specify more information
Yes	3	The website should index at least 1000 documents;
Yes	4	The search engine accepts a text query in the search box
Yes	5	The search engine should return search results on a separate page
Yes	6	The search engine can prevent XSS vulnerability by removing tags existing in the query
Yes	7	The search engine result page (SERP) should contain a search box
No	8	The advanced search should return results satisfying multiple specifications
Yes	9	Users should be able to insert a new entry and search engine will index it.

2. Search Index Schema Design

Describe how index schema is designed. For example, for the account.json file, used as an example in class, the schema design can be shown as a table below. The index name is “account”.

Table 2: Search index schema design.

#	Field	Type	Example
1	Account_number	Int	1
2	Balance	Int	39225
3	Firstname	String	Amber
4	Lastname	String	1990
5	Age	Int	32
6	Gender	String	M

7	Address	String	880 Holmes Lane
8	Employer	String	Pyrami
9	Email	String	amberduke@pyrami.com
10	City	String	Brogan
11	State	String	IL

3. Search Function

```php
<?php
require 'vendor/autoload.php';
use Elasticsearch\ClientBuilder;

$hosts = [
    'localhost:9200'
];

$client = ClientBuilder::create()->setHosts($hosts)->build();
```

The search function is implemented by using Elastic search and Php together along with some JavaScript. The image shows the important line of code used to connect the frontend pages to the Elasticsearch. We should make sure to define the host along with the port number (9200 by default). We should use the composer to connect the php to the Elastic search. We have to include elasticsearch-php in our composer json file. Composer will also create the vendor/autoload.php file that is a requirement for the php to run with Elastic search without any issues.

There was a requirement for our search engine to prevent XSS vulnerability by removing tags existing in the query. This requirement Is achieved by using the simple strip_tags() function to eliminate the tags existing while the user input any keyword for searching. I have implemented an additional step to avoid cross - site scripting attacks by using the trim() function to validate the input to be a non zero length string . Then the search term is sanitized by removing any html tags it may contain. Also, while displaying the results I have made sure to filter the output by using htmlspecialchars() function. These simple steps that were taken will avoid any XSS vulnerability attacks by any user.

```php
$searchterm ="" ;

if(isset($_GET["keyword"]))
    #validate searchterm
    $searchterm = trim($_GET['keyword']);
    #sanitize searchterm
    $searchterm = strip_tags($searchterm);
```

```php
else{
    $params = [
    'index' => 'bank',
    'size'   => 1000,
      'body' => [
         'query' => [
            'bool' => [
               'should' => [

                  ['match' => ['firstname' => "{$searchterm}"]],
                  ['match' => ['lastname' => "{$searchterm}"]],
                  ['match' => ['gender' => "{$searchterm}"]],
                  ['match' => ['address' => "{$searchterm}"]],
                  ['match' => ['employer' => "{$searchterm}"]],
                  ['match' => ['email' => "{$searchterm}"]],
                  ['match' => ['city' => "{$searchterm}"]],
                  ['match' => ['state' => "{$searchterm}"]]
               ]
            ]
         ],
      ]
   ];
$response = $client->search($params);
```

The matching of attributes was done by using the Boolean function and should match option. Once the matching has been completed the search results were saved into a variable called $response which is of type array. Unless otherwise the search results are converted to a more user-friendly way, the results will be displayed as an array. We can use the print_r function to directly print the output and view it through our browser. However, I have taken several steps to make sure the search results will be displayed in a separate page (SERP) in a more user- readable way than just an array.

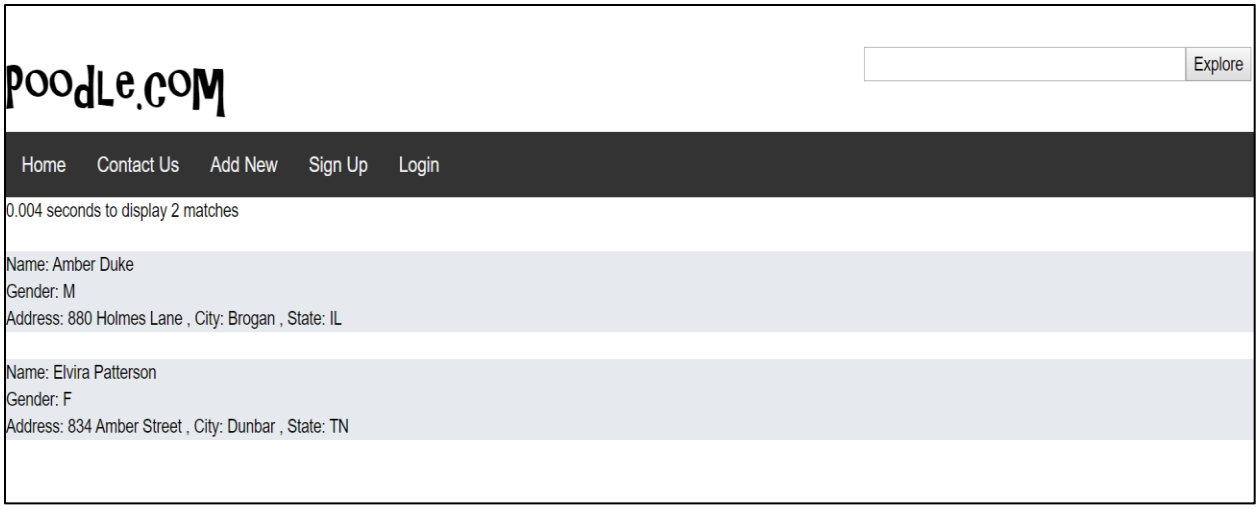

Figure 1: SERP

4. Challenges and Lessons.

During the process of achieving the requirements in this milestone, I have faced several challenges in each stage. The most challenging part of this Milestone was to find an appropriate dataset which satisfy all of my requirements. The datasets about dogs were available only towards a more specific area (intelligence, registered names, etc.) It was not possible to find a dataset with all the characteristics of a dog. I used an alternate dataset (accounts.json) to work out the features which was provided as an example in class. Finally, decided to use available datasets along with some fake data added manually to make sure the dataset is complete.

It was also quite challenging to figure out how to connect the php with the Elasticsearch. By revisiting the lecture slides and checking online sources it was easier to overcome that challenge. Also, it was very challenging to get the steps in order to display the search results in a more user-friendly way after you get the search results as an array. No clear reference sources were available on it either. Upon trying different approaches, I managed to overcome that challenge as well.

During this milestone I learned that the display of data is not an easy task as it may look like. I would spend more time on improving the design to make sure the search results page will resemble most popular search engine result pages (Google, Bing, etc.).

5. Additional features

Firstly, I would make sure that the real dataset is used and all the features implemented till now are in working stage. The re-captcha to verify that the user is a human not a bot or web crawler is in developing stage. The planned future work is to complete the re-captcha verification process completed. I have done research on how the paginated results should be returned where each match will have a single page for dedicated for it. Also, I am planning to give full attention to make improvements to the front-end design. This will be done in a way that the search engine's look and feel to be better and user-friendly.