

Техническое задание на разработку сервиса проверки позиций сайтов по ключевым запросам

1. Цель работ

Сделать кастомную панельку, отображающую позиции наших сайтов в Google по заданным ключевым запросам.

2. Состав работ

Нужно сделать:

- Админку, работающую в веб-браузере
- Скрипт проверки позиций сайтов, забитых в админку в Google по ключам, также из админки

2.1. Админка

Нужно сделать админку со следующими разделами (страницами):

- Вход по логину-паролю
- Настройки
- Пользователи
- Карточка пользователя
- Проекты
- Карточка проекта
- Позиции
- Позиции по проекту
- Позиции по сайту
- Позиции по ключу
- Собрать данные за сегодня

Админку сделать на каком-то нормальном шаблоне, заранее согласовать примерный вид.

Общая верстка админки такая:

- Слева узкий сайдбар
- Справа область контента

В сайдбаре нужно меню (которое появляется после успешного логина), каждый пункт которого - ссылка:

- Позиции – переход в раздел Позиции
- с отступом в пару пробелов вправо – список проектов, один проект на строке, чтобы можно было кликнуть сразу ссылку и перейти в Позиции по этому проекту
- с отступом 1 строку вниз Проекты
- с отступом в пару пробелов вправо - список проектов, один проект на строке, чтобы можно было кликнуть сразу ссылку и перейти в Карточку этого проекта
- с отступом 1 строку вниз Собрать данные за сегодня
- с отступом 1 строку вниз Пользователи
- с отступом 1 строку вниз Настройки
- с отступом 1 строку вниз Выход – разлогин и возврат на страницу логина.

Меню в сайдбаре всегда развернуто. Никакого сворачивания/разворачивания не надо. Скролла не надо.

Требования к базе данных: MySQL

- таблицы должны начинаться с колонки `int AUTO_INCREMENT PRIMARYKEY`
- по таблицам Sites, Keys должны быть созданы индексы по коду проекта
- все даты хранить типом данных `int` в формате YYYYMMDD, т.е. дату 21.01.2020 храним в базе как целое число 20200120

Сама админка на PHP, фреймворк по желанию.

2.1. Вход по логину паролю

Окно входа по логину паролю. Логин будет вида "misha". Функций восстановления забытого пароля с отправкой на емейл и т.п. не надо.

Разделения по ролям нет или разграничения доступа нет. Все юзеры имеют доступ ко всем функциям (= админы) и ко всем данным.

После успешного логина переходим на страницу Позиции.

2.2. Настройки

Настройки это страница с настройками и кнопкой Сохранить внизу.

Пока будут следующие настройки:

- URL XMLPROXY – пример значения
<http://xmlstock.com/google/?user=1&key=15f48712d04be797df3950d2fd75271>

Все настройки делать в виде лейбла: инпут

Внизу кнопка Сохранить, по которой сохраняем и возвращаем в раздел Настройки

В базе под настройки завести таблицу Settings с колонками `setting_id`, `setting_name varchar(50)`, `setting_value varchar(250)`

2.3. Пользователи

Это реестр пользователей, в начале их будет 3, потом может еще пара. Оформить в виде таблицы с заголовками колонок

Логин	Проекты	Действия
-------	---------	----------

Логин – это логин пользователя, например, misha

Проекты – это перечисление проектов, которые у пользователя отображаются в сайдбаре (разделы Проекты и Позиции) и в самих этих разделах Проекты и Позиции. Т.е. это те проекты, с которыми работает пользователь. Перечисление через пробел, например,

site1.ru site222.com gamesforgirls.online b21aaaa.xyz

Действия – это две ссылки Редактировать и Удалить

По ссылке Редактировать переходим в карточку пользователя

По ссылке Удалить спрашиваем подтверждение, если да – удаляем пользователя. Удаление пользователя не удаляет проекты, но удаляет строки в таблице, в которой храним список проектов, с которыми работает пользователей (чтобы не было зависших строк в базе).

В базе под юзеров завести таблицу Users с колонками user_id, user_name varchar(50) и таблицу UserProjects row_id, user_id, project_id – эта таблица для хранения ID шников проектов, с которыми работает юзер.

2.4. Карточка пользователя

Это страничка редактирования пользователя. Здесь можно поменять логин, задать пароль и указать проекты, с которыми пользователь будет работать.

Выглядит так:

Логин: [misha] – логин не пустой

Пароль: [] - пароль не выводим, храним хеш, если что-то ввели в поле – меняем пароль в базе, если поле не заполнили – не меняем. При добавлении пользователя требовать заполнения логина и пароля.

Проекты: - здесь набор чекбоксов с названиями проектов. Если пользователь работает с проектом – в чекбоксе галка, если не работает – без галки.

Внизу кнопка Сохранить, по которой возврат в раздел Пользователи

2.5. Проекты

Это таблица со списком проектов. Выводим в виде таблицы со следующими колонками:

Сайты и ключи вводятся /редактируются в текстарях. При сохранении карточки проекта удаляем строки из таблиц ProjectSites и ProjectKeys по этому проекту и добавляем их заново если юзер заполнил их в текстарях.

Ключи и сайты вводятся в текстарях по одному на строку или через запятую. Поэтому при сохранении сначала заменять запятую на перенос строки, превращать в массив и непустые элементы сохранять в базу.

Сайты могут заводиться с http://, https:// с www. или без, поэтому при сохранении приводить все к названию домена (он может быть и третьего уровня), т.е. отрезать http://, https://, www. заменяя на пустую строку.

Примеры (сайты)

<https://gamesites.com> -> gamesites.com

<http://www.site.ru> -> site.ru

www.games.online -> games.online

при сохранении обновляем у проекта значения project_sites и project_keys (считая кол-во сайтов и ключей в проекте).

2.7. Позиции

В этом разделе в дальнейшем будет сводная информация по позициям в целом, пока непонятно как ее готовить и выводить, поэтому при заходе в этот раздел выводим в контенте

Выберите проект: ссылки на проекты

ссылки на проекты – набор ссылок на страницу “Позиции по проекту” по проектам, с которыми работает юзер, например,

Выберите проект: [Игры для девочек](#) [Аркады](#) [Гонки на мотоциклах](#) [.io игры](#)

В базе под позиции делаем таблицу Positions с колонками

- row_id int
- pos_date int
- key_name varchar(200)
- pos_num int
- domain_name varchar(200)

2.8. Позиции по проекту

В этом разделе выводим список позиций сайтов проекта по ключам проекта в Гугле в виде:

Заголовок “Название проекта”, например, Игры для девочек

Таблица с колонками:

- Сайт – domain_name, например, gamesites.com
- Ключ – key_name, например, play girl games

- Дата – в формате DD.MM.YYYY, например, 21.01.2020

Дат будет много, т.е. например,

Сайт	Ключ	20.01.2020	12.01.2020	01.01.2020
gamesites.com	play girl games		5	43

важно – если позиции на эту дату по этому ключу и сайту нет, то выводим пробел () в ячейку – как в примере выше 01.01.2020 ключа еще не было в выдаче гугла.

Таблицу выводим целиком, без пагинатора. Но с горизонтальным скроллом.

Формируем таблицу следующим образом: делаем SQL запрос вида

```
SELECT P.domain_name, P.key_name, P.pos_date, P.pos_num FROM Positions P WHERE
P.domain_name IN (SELECT domain_name FROM ProjectSites WHERE ProjectSites.project_id = %PID%)
AND P.key_name IN (SELECT key_name FROM ProjectKeys WHERE ProjectKeys.project_id = %PID)
ORDER BY P.pos_date DESC, P.pos_num
```

Т.е. получаем все позиции, которые есть в базе по всем ключам и всем сайтам данного проекта за все время. Далее этот датасет сохраняем в массив arrData и делаем несколько прогонов для набивки таблицы:

- В первый прогон считаем сколько у нас разных дат и сколько у нас разных комбинаций domain_name.key_name и делаем в памяти двумерный массив arrTable размера ширина “Кол-во дат” + 2, высота кол-во уникальных комбинаций domain_name.key_name. Когда считаем разные даты набиваем массив arrDates разными датами (в примере выше будет одномерный массив из трех значений 20200120, 20200112 и 20200101).
- Второй прогон набиваем arrTable данными пробега по массиву arrData и обрабатывая строки массива следующим образом:
 - Делаем цикл от 0 до размера в высоту arrTable и пробегаем по строкам arrTable ищем есть ли строка с доменом и ключом из arrData. Если встречаем в arrTable незаполненное название сайта (т.е. дошли до конца заполненной области arrTable) – выходим из цикла
 - Если строка НЕ нашлась (т.е. еще в arrTable нет данных по этому сайту и ключу), то вносим в arrTable сайт и ключ. После чего пробегаем по массиву arrDates и смотрим какой индекс в этом массиве у нашей даты, например, у даты 20200112 из примера ниже будет индекс 1 (у 20200120 – 0, у 20200101 – 2). После чего пишем в arrTable[номер строки, 2+ номер индекса даты] значение позиции по данному сайту и ключу в данный день
 - Если строка нашлась, то аналогично вычисляем какая колонка соответствует дате и пишем туда значение позиции.

Т.е. после второго прогона у нас будет в arrTable что-то такое

gamesites.com	play girl games	5	43	пусто
gamesites.com	barbie games	пусто	29	пусто
somesite.com	play online	99	25	25

т.е. у нас в памяти массив, который содержит ячейки из таблицы, которую нужно показать – ее генерируем, собирая <table> <tr> и <td> и выводим на странице.

Название сайтов делаем невидимым ссылками (т.е. выглядят как обычный текст без другого цвета и подчеркивания, но при наведении подчеркивание появляется и в верстке это именно <a) на страницу “Позиции по сайту”

Аналогично ключи – ссылки на страницу “Позиции по ключу”

2.9. Позиции по сайту

Позиции по сайту – аналогично позициям по проекту (таблица без колонки Сайт), только если в позициях по проекту мы выводим все сайты проекта и название проекта в заголовке, то в Позициях по сайту выводим в заголовке название сайта и позиции только по этому сайту, например, по gamesites.com

Название сайта можно передавать через GET параметр в ссылке, а дальше в php выбирать site_id и project_id из ProjectSites WHERE domain_name = \$_GET[“domain_name”]

```
SELECT P.key_name, P.pos_date, P.pos_num FROM Positions P WHERE P.domain_name = (SELECT domain_name FROM ProjectSites WHERE ProjectSites.site_id = %SID%) AND P.key_name IN (SELECT key_name FROM ProjectKeys WHERE ProjectKeys.project_id = %PID) ORDER BY P.pos_date DESC, P.pos_num
```

2.10. Позиции по ключу

Позиции по ключу – это таблица какие сайты были на каких позициях по данному ключу. Сверху заголовков Позиции по ключу [key_name]:

дальше таблица, колонки в таблице это Позиция и даты, например,

Позиция	20.01.2020	12.01.2020	01.01.2020
1	site1.ru	site2.ru	site3.ru
2	site2.ru	site1.ru	site5.com
3	site8.com	gamesforgirls.com	arcadaonline.ru

Ключ можно передавать через GET параметр в ссылке.

Запрос для выборки данных такой

```
SELECT P.domain_name, P.pos_date, P.pos_num FROM Positions P WHERE P.key_name = %KNAME% ORDER BY P.pos_date DESC, P.pos_num
```

2.11. Собрать данные за сегодня

Эта страница предназначена для ручного запуска сбора данных и контроля прогресса по сбору данных, должна содержать:

- Заголовок Сбор данных за %дата%
- Информация по данным
- Кнопка Запустить сбор данных

Блок “Информация по данным” заполняем следующим образом:

- Делаем запрос `SELECT COUNT(*) FROM Positions WHERE pos_date = %текущаядата%`, сохраняем в `$total`
- Если `$total == 0` – выводим “данных нет”
- Если `$total != 0`, то делаем запрос

```
SELECT COUNT(*) FROM Positions WHERE pos_date = %текущаядата% AND domain_name IS NULL
```

Сохраняем в `$todo`

Если `$todo == 0` то выводим “сбор данных успешно завершен.”

Если `$todo != 0`, то выводим “сбор данных в процессе, выполнено на X%”, где $X = \text{floor}(\$todo * 100 / \$total)$.

По Кнопке сбор данных делаем следующее:

- Запрашиваем подтверждение “Вы действительно хотите собрать данные за %DATE%?”. Если нет – ничего не делаем.
- Если подтверждают, то

```
DELETE FROM Positions WHERE pos_date = %текущаядата%
```

`SELECT DISTINCT key_name FROM ProjectKeys` – получим список всех ключей, который надо мониторить. Делаем цикл и по каждому ключу делаем внутренним циклом пакетный запрос на вставку в базу данных заглушек, в которые парсер будет писать позиции. По каждому ключу делаем цикл `$i` от 1 до 100, который в один `$query` собирает INSERTы в Positions и выполняет их скопом (за один запрос к mysql)

```
INSERT INTO Positions(pos_date, key_name, pos_num, domain_name) VALUES(%ТЕКУЩАЯДАТА%, %КЛЮЧ%, $i, “);
```

Т.е. например,

```
INSERT INTO Positions(pos_date, key_name, pos_num, domain_name) VALUES(20200121, ‘play online’, 1, NULL);
```

```
INSERT INTO Positions(pos_date, key_name, pos_num, domain_name) VALUES(20200121, ‘play online’, 2, NULL);
```

```
INSERT INTO Positions(pos_date, key_name, pos_num, domain_name) VALUES(20200121, ‘play online’, 3, NULL);
```

...

```
INSERT INTO Positions(pos_date, key_name, pos_num, domain_name) VALUES(20200121, ‘play online’, 100, NULL);
```

После чего выводим пользователю сообщение “Процесс сбора данных запущен, ожидайте его завершения.”.

2.12.Скрипт сбора данных

Скрипт сбора данных нужно реализовать в виде запускающегося каждую минуту по крону консольного php скрипта, выполняющего следующее: проверку работает ли сбор данных, если не работает – выполнение сбора данных.

В ходе сбора данных скрипт должен писать в ту же папку, где он расположен файл-маркер с 1 внутри. Выполняя запись он обновляет дату файла.

При запуске скрипта проверяем дату файла-маркера, если файл отсутствует, значит сбор данных еще не осуществлялся и его можно провести.

Если файл-маркер присутствует, но дата отличается от текущей \geq чем на 10 минут, то прошлый сбор данных завис и можно начинать новый сбор данных.

Если файл-маркер присутствует, но дата отличается от текущей больше меньше чем на 10 минут, то сбор данных идет, из этого экземпляра скрипта выходим.

Сам сбор данных выполняем следующим образом:

- 1) Выбираем ключи для сбора данных

```
SELECT DISTINCT key_name FROM Positions WHERE domain_name IS NULL
```

- 2) По каждому ключу дергаем xmlstock.com (см. url в настройках) и парсим выдачу гугла по данному ключу, сохраняя результаты в базу.

Хелп здесь <http://xmlstock.com/?do=google-help>

В базу сохраняем аналогично единым sql запросом сразу 100 позиций

```
UPDATE Positions SET domain_name = %SITE1% WHERE key_name = %KEY_NAME% AND domain_name IS NULL AND pos_num = 1;
```

```
UPDATE Positions SET domain_name = %SITE2% WHERE key_name = %KEY_NAME% AND domain_name IS NULL AND pos_num = 2;
```

```
UPDATE Positions SET domain_name = %SITE3% WHERE key_name = %KEY_NAME% AND domain_name IS NULL AND pos_num = 3;
```

...

```
UPDATE Positions SET domain_name = %SITE100% WHERE key_name = %KEY_NAME% AND domain_name IS NULL AND pos_num = 100;
```

Если xmlstock.com (гугл) выдал меньше 100 ответов на поиск по ключу, то сохраняем не выданные до 100 пустыми значениями. Скажем гугл на поиск по ключу “какой то редкий ключ никто не ищет такое” выдал 5 ответов, значит сохраняем 100 строк, из которых у тех, что pos_num с 1 по 5 будут domain_name заполнены, а у 6-100 будет “ (пустая строка, не NULL)

После каждого пакетного апдейта обновляем файл-маркер.

СПАСИБО, ЧТО ПОСМОТРЕЛИ ЭТО ТЗ ДО КОНЦА. В ОТВЕТЕ НА ПРОЕКТ ПОЖАЛУЙСТА УКАЖИТЕ 111, ЧТОБЫ Я ЭТО ПОНЯЛ 😊