

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГБОУ ВО «Тувинский государственный университет»
ФИЗИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ
КАФЕДРА ИНФОРМАТИКИ**

Выпускная квалификационная работа
(бакалаврская работа)

**WEB-ПРИЛОЖЕНИЕ ДЛЯ СРАВНЕНИЯ ЦЕН В КАФЕ ГОРОДА
КЫЗЫЛ**

Работа допущена к защите
И.о.зав. кафедрой информатики
Тарыма А.К., к.п.н., доцент
(фамилия, и.о., уч.степень, звание)

Студента 4 курса 4 группы
направления подготовки 02.03.02
Фундаментальная информатика и
информационные технологии
Суван Айдыс Викторович

Работа защищена «19» июня 2020г.
С оценкой _____
Председатель ГЭК _____
(подпись)

(подпись)
«__» _____ 2020 г.

Сенашов В.И. д.ф-м.н., профессор
ведущий научный сотрудник
Института
вычислительного моделирования
СО РАН, Красноярск

Члены комиссии _____

(подписи)

Научный руководитель:

(подпись)
Монгуш Ч.М., ст. преп. кафедры
информатики., к.ф.-м.н.
(фамилия, и. о., должность, уч.степень, звание)

Кызыл – 2020 г.

Содержание

Введение.....	3
Глава 1. Программные средства для создания web-приложения.....	5
1.1. Локальный web-сервер OpenServer.....	5
1.2. Система управления базами данных MySQL.....	7
1.3. Язык разметки гипертекста HTML..	8
1.4. Каскадные таблицы стилей CSS.....	9
1.5. Скриптовый язык программирования PHP..	10
1.6. Библиотека phpQuery.....	12
1.7. Фреймворк Bootstrap.....	12
1.8. Среда разработки Sublime Text 3.....	13
Глава 2. Web-приложение для сравнения цен в кафе города КЫЗЫЛ	14
2.1. Структура базы данных.....	14
2.1. Структура web-приложения.....	16
2.2. Использование PHP, phpQuery и MySQL для парсинга информации с сайтов кафе.....	23
Заключение	24
Список использованной литературы.....	25
Приложение 1	26
Приложение 2	40
Приложение 3	49
Приложение 4	58

Введение

В сети Интернет можно увидеть множество сервисов сравнения цен (прайс-агрегаторы) на различные продукты и товары, предлагаемых интернет-магазинами. Основные из них: Shops-Prices, trivago, E-Katalog, Яндекс Маркет, Price. Такие сервисы предназначены для того, чтобы помочь покупателям найти товары по лучшей цене. Кроме сравнения цен в этих сервисах пользователь может прочитать отзывы других или же оставить свой, а также получить информацию о товаре.

С каждым годом становится все больше людей, которые пользуются прайс-агрегаторами. Так как, во-первых, с их помощью легко найти самые выгодные предложения. Во-вторых, они очень удобны для использования, поскольку покупатель видит сразу на одной странице все предложения по выбранному его товару среди множества интернет-магазинов. Данные товары сортируются от самого дешёвого до самого дорогого.

Актуальность создания web-приложения для сравнения цен в кафе города Кызыл заключается в том, что он предоставляет пользователям получить свежую информацию о блюдах и напитках в кафе города Кызыл на одном сайте, которая будет обновляться ежедневно. Для разработки web-приложения выбираются кафе города Кызыл с действующим сайтом.

Целью выпускной квалификационной работы является разработка web-приложения для сравнения цен в кафе города Кызыл с помощью языка программирования PHP с использованием базы данных MySQL и библиотеки phpQuery.

Для достижения поставленной цели необходимо решить следующие **задачи**:

- изучить литературу по web-программированию;
- спроектировать и создать базу данных для web-приложения, где будут храниться данные о кафе и товарах;

- подключить библиотеку phpQuery к web-приложению для парсинга данных;
- создать панель администратора, позволяющая администратору добавлять, редактировать и удалять записи о кафе и о еде;
- написать скрипт, который сортирует товары (блюда) по цене;
- разработать web-приложение для сравнения цен в кафе города Кызыл.

Объектом исследования данной работы является разработка программного продукта.

Предметом исследования создание web-приложения для сравнения цен в кафе города Кызыл, а также его административной панели.

Новизна работы заключается в том, что web-приложение для сравнения цен в кафе разработан впервые по городу Кызыл и на момент создания не имеет подобных программных продуктов.

Практическая значимость работы состоит в том, что приложение для сравнения цен на еду в кафе будет использоваться жителями и гостями города Кызыла, а также позволит конкурентам вести мониторинг цен.

Глава 1. Программные средства для создания web-приложения

1.1. Локальный web-сервер OpenServer

Web-приложение разрабатывалось на языке PHP, а web-страницы на языке HTML с использованием CSS и Bootstrap 4 – это открытый и бесплатный HTML, CSS и JS фреймворк, который используется веб-разработчиками для быстрой вёрстки адаптивных дизайнов сайтов и веб-приложений. Для работы PHP требуется установить локальный web-сервер.

Веб-сервер – сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, как правило, вместе с HTML-страницей, изображением, файлом, медиа-поток или другими данными.

Веб-сервером называют как программное обеспечение, выполняющее функции веб-сервера, так и непосредственно компьютер, на котором это программное обеспечение работает.

Работает сервер следующим образом: клиент запрашивает документ при помощи URL; web-сервер принимает запрос, находит в системе URL, соответствующий заданному запросу, – физический файл (который может быть файлом HTML или файлом другого типа). Убедившись, что клиент имеет разрешение на получение файла, web-сервер возвращает файл клиенту. Кроме того, обычно web-сервер ведет запись протокола: кто и какой документ запрашивал, и сколько было запросов, что позволяет получить статистику для определения популярности web-сайта.

Наиболее известными и распространёнными web-серверами являются:

- Apache – примерно 40.80% всех сайтов в Интернете;
- Nginx – примерно 23.13% всех сайтов в Интернете;
- Microsoft-IIS – примерно 17.28% всех сайтов в Интернете.

Apache HTTP-сервер (сокращение от англ. Apache server) – свободный веб-сервер. Apache является кроссплатформенным ПО, поддерживает

операционные системы Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS.

Основными достоинствами Apache считаются надёжность и гибкость конфигурации. Он позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках и т. д. Поддерживает IPv6. Наиболее вескими недостатками являются:

- возможные проблемы с производительностью на высоконагруженных сайтах с большим трафиком, «тяжелым» контентом или приложениями, которые требуют высоких вычислительных мощностей;
- большое количество параметров настройки может привести к возникновению уязвимостей, из-за невнимательности при конфигурации;
- существует вероятность того, что в модули от независимых разработчиков будет внедрен вредоносный код.

В качестве локального web-сервера использовалось Open Server.

Open Server – локальный WAMP/WNMP сервер, работающий как в стационарном, так и в портативном режиме, имеющий многофункциональную управляющую программу и большой выбор подключаемых компонентов.

Состав программного комплекса

Основные компоненты:

- OSPanel 5.3.5
- Apache 2.2.31 / 2.4.38 / 2.4.41
- Bind 9.14.5
- ConEmu 19.07.14
- FTP FileZilla 0.9.60
- Ghostscript 9.27
- Git 2.23.0
- HeidiSQL 10.2.0.5599

- Nginx 1.17.3
- NNCron Lite 1.17
- Sendmail 32
- Wget 1.20.3

Системы управления базами данных:

- MariaDB 5.5.63 / 10.0.38 / 10.1.38 / 10.2.22 / 10.3.13
- Memcached 1.2.6 / 1.4.5
- MongoDB 2.4.14 / 2.6.12 / 3.0.15 / 3.2.22 / 3.4.19 / 3.6.11 / 4.0.6 / 4.2.0
- MySQL 5.1.73 / 5.5.62 / 5.6.43 / 5.7.25 / 8.0.15
- PostgreSQL 9.2.24 / 9.3.25 / 9.4.21 / 9.5.16 / 9.6.12 / 10.7 / 11.2
- Redis 2.8.2402 / 3.0.504 / 3.2.100
- PHP модули:
- PHP 5.2.17 - PHP 7.3.9 + расширения

PHP приложения:

- Adminer 4.7.3
- PHPMemcachedAdmin 1.3
- PHPMyAdmin 4.9.0.1
- PHPPgAdmin 7
- PHPRedisAdmin 1.11.4

1.2. Система управления базами данных MySQL

MySQL – это одна из самых популярных и самых распространенных систем управления базами данных (СУБД) в интернете. Она не предназначена для работы с большими объемами информации, но ее применение идеально подходит интернет сайтов, как небольших, так и достаточно крупных.

MySQL отличается тем что, работает на высокой скорости, надежностью и гибкостью. Работа с ней, как правило, не вызывает больших трудностей. Поддержка сервера MySQL автоматически включается в поставку PHP. Немаловажным фактором является ее бесплатность. MySQL распространяется на условиях общей лицензии GNU (GPL, GNU Public License).

Ранее для длительного хранения информации разработчики работали с файлами: помещали в них некоторое количество строчек, а затем извлекали их для последующей работы. Задача длительного хранения информации очень часто встречается в программировании Web-приложений: подсчёт посетителей в счётчике, хранение сообщений в форуме, удалённое управление содержанием информации на сайте и т.д.

Все эти проблемы решает использование базы данных. Базы данных сами заботятся о безопасности информации, и её сортировке, которые позволяют извлекать и размещать информацию при помощи одной строки. Код с использованием базы данных получается более компактным, и отлаживать его гораздо легче. Кроме того, не нужно забывать и о скорости - выборка информации из базы данных происходит значительно быстрее, чем из файлов.

1.3. Язык разметки гипертекста HTML

HTML (от англ. HyperText Markup Language – «язык разметки гипертекста») – стандартный язык разметки документов во Всемирной паутине. Большинство веб-страниц создаются при помощи языка HTML. Язык HTML интерпретируется браузером и отображается в виде документа, в удобной для человека форме.

HTML является приложением SGML (стандартного обобщённого языка разметки) и соответствует международному стандарту ISO 8879.

Язык HTML был разработан британским учёным Тимом Бернерсом-Ли приблизительно в 1991–1992 годах в стенах Европейского совета по ядерным исследованиям в Женеве (Швейцария). HTML создавался как язык для обмена научной и технической документацией, пригодный для использования людьми, не являющимися специалистами в области вёрстки. HTML успешно справлялся с проблемой сложности SGML путём определения небольшого набора структурных и семантических элементов – дескрипторов. Дескрипторы также часто называют «тегами». С помощью HTML можно легко создать относительно простой, но красиво оформленный документ. Помимо упрощения структуры документа, в HTML внесена поддержка гипертекста. Мультимедийные возможности были добавлены позже. Изначально язык HTML был задуман и создан как средство структурирования и форматирования документов без их привязки к средствам воспроизведения (отображения). В идеале, текст с разметкой HTML должен был без стилистических и структурных искажений воспроизводиться на оборудовании с различной технической оснащённостью (цветной экран современного компьютера, монохромный экран органайзера, ограниченный по размерам экран мобильного телефона или устройства и программы голосового воспроизведения текстов).

1.4. Каскадные таблицы стилей CSS

Для придания красивого дизайна системы управления базами данных используются каскадные таблицы стилей.

CSS (англ. Cascading Style Sheets – каскадные таблицы стилей) – технология описания внешнего вида документа, написанного языком разметки. CSS – одна из широкого спектра технологий, одобренных консорциумом W3C и получивших общее название «стандарты Web». В 90-х годах стала ясна необходимость стандартизировать Web, создать какие-то

единые правила, по которым программисты и веб-дизайнеры проектировали бы сайты. Так появились языки HTML 4.01 и XHTML и стандарт CSS.

В начале 90-х, различные браузеры имели свои стили для отображения веб-страниц. HTML развивался очень быстро и был способен удовлетворить все существовавшие на тот момент потребности по оформлению информации, поэтому CSS не получил тогда широкого признания.

Термин «каскадные таблицы стилей» был предложен Хокон Виум Ли в 1994 году. Совместно с Бертом Босом он стал развивать CSS.

В отличие от многих существовавших на тот момент языков стиля, CSS использует наследование от родителя к потомку, поэтому разработчик может определить разные стили, основываясь на уже определенных ранее стилях. В середине 90х Консорциум Всемирной Паутины (W3C) стал проявлять интерес к CSS, и в декабре 1996 года была издана рекомендация CSS1.

Конкретные преимущества CSS:

- управление отображением множества документов с помощью одной таблицы стилей;
- более точный контроль над внешним видом страниц;
- различные представления для разных носителей информации (экран, печать, и т. д.);
- сложная и проработанная техника дизайна.

1.5. Скриптовый язык программирования PHP

PHP (англ. PHP: Hypertext Preprocessor «PHP: препроцессор гипертекста»; первоначально Personal Home Page Tools – «Инструменты для создания персональных веб-страниц») – скриптовый язык программирования общего назначения, интенсивно применяемый для разработки web-приложений.

Значительным отличием PHP от какого-либо кода, выполняющегося на стороне клиента, например, JavaScript, является то, что PHP-скрипты

выполняются на стороне сервера. Вы даже можете сконфигурировать свой сервер таким образом, чтобы HTML-файлы обрабатывались процессором PHP, так что клиенты даже не смогут узнать, получают ли они обычный HTML-файл или результат выполнения скрипта.

PHP позволяет создавать качественные Web-приложения за очень короткие сроки, получая продукты, легко модифицируемые и поддерживаемые в будущем.

PHP прост для освоения, и вместе с тем способен удовлетворить запросы профессиональных программистов.

Расширение Объекты данных PHP (PDO) определяет простой и согласованный интерфейс для доступа к базам данных в PHP. Каждый драйвер базы данных, в котором реализован этот интерфейс, может представить специфичный для базы данных функционал в виде стандартных функций расширения. Но надо заметить, что само по себе расширение PDO не позволяет манипулировать доступом к базе данных. Чтобы воспользоваться возможностями PDO, необходимо использовать соответствующий конкретной базе данных PDO драйвер.

PDO обеспечивает абстракцию (доступа к данным). Это значит, что вне зависимости от того, какая конкретная база данных используется, вы можете пользоваться одними и теми функциями для выполнения запросов и выборки данных. PDO не абстрагирует саму базу данных, это расширение не переписывает SQL запросы и не эмулирует отсутствующий в СУБД функционал. Если нужно именно это, необходимо воспользоваться полноценной абстракцией базы данных.

Расширение PDO внедрено в PHP 5.1, но также доступно в 5.0 в виде PECL расширения; PDO использует новый ОО функционал из ядра PHP 5, соответственно оно не будет работать с ранними версиями PHP.

1.6. Библиотека phpQuery

phpQuery – это цепочечный API-интерфейс на основе объектной модели документов (DOM), управляемый селектором на стороне сервера, на основе библиотеки JavaScript jQuery.

phpQuery – это библиотека для парсинга сайтов.

Большинство функциональных возможностей, предоставляемых jQuery в phpQuery, в целом можно разделить на 4 задачи:

- создание элементов DOM;
- выбор и работа с элементами;
- перебор элементов DOM;
- вывод в браузер.

Вы можете выполнить задачи, используя возможности, предоставляемые phpQuery, который известен как «ported jQuery sections»:

- **Selectors** — найти элементы на основе заданных условий.
- **Attributes** — работа с атрибутами элементов DOM.
- **Traversing** — путешествие через список выбранных элементов.
- **Manipulation** — добавление и удаление содержимого выбранных элементов.
- **Ajax** — создание сервера Ajax запросов.
- **Events** — связать DOM событий на отдельных элементах.

1.7. Фреймворк Bootstrap

Bootstrap – это открытый и бесплатный HTML, CSS и JS фреймворк, который используется веб-разработчиками для быстрой вёрстки адаптивных дизайнов сайтов и веб-приложений.

Фреймворк Bootstrap используется по всему миру не только независимыми разработчиками, но иногда и целыми компаниями. На

Bootstrap создано очень много различных сайтов, посмотреть на них можно на странице Bootstrap Expo.

Основная область его применения – это разработка frontend сайтов и интерфейсов админок. Среди аналогичных систем (Foundation, UIKit, Semantic UI, InK и др.) фреймворк Bootstrap является самым популярным.

Почему Bootstrap так популярен? Это связано с тем, что он позволяет верстать сайты в несколько раз быстрее, чем это можно выполнить на «чистом» CSS и JavaScript. А в нашем мире время – это самый дорогой ресурс. Также его популярность связана с доступностью. Она заключается в том, что на нём даже начинающий разработчик может верстать достаточно качественные макеты, которые трудно было бы выполнить без глубоких знаний веб-технологий и достаточной практики.

Фреймворк Bootstrap представляет собой набор CSS и JavaScript файлов. Чтобы его использовать на странице эти файлы нужно просто подключить к ней. После подключения вам станут доступны инструменты данного фреймворка: колоночная система (сетка Bootstrap), классы и компоненты.

1.8. Текстовый редактор Sublime Text 3

Sublime Text Editor – это полнофункциональный текстовый редактор для редактирования локальных файлов или базы кода. Он включает в себя различные функции для редактирования базы кода, которая помогает разработчикам отслеживать изменения. Различные функции, которые поддерживаются в Sublime Text: подсветка синтаксиса; авто отступ; распознавание типов файлов; боковая панель с файлами указанного каталога; макрос; плагины и пакеты.

Sublime Text Editor используется, как интегрированный редактор разработки (IDE), как код Visual Studio и NetBeans. Текущая версия редактора Sublime Text – 3.0 и совместима с различными операционными системами, такими как Windows, Linux и MacOS.

Глава 2. Web-приложение для сравнения цен в кафе города Кызыл

2.1. Структура базы данных

База данных web-приложения для сравнения цен в кафе города Кызыл состоит из 5 таблиц:

safe – таблица (рис.1) содержит данные о кафе.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id 	int(11)		UNSIGNED	Нет	Нem		AUTO_INCREMENT
2	cafe_title	varchar(255)	utf8_general_ci		Нет	Нem		
3	cafe_url	varchar(255)	utf8_general_ci		Нет	Нem		
4	address	varchar(255)	utf8_general_ci		Да	NULL		
5	phone	varchar(100)	utf8_general_ci		Да	NULL		
6	cafe_img	varchar(255)	utf8_general_ci		Да	NULL		
7	cafe_about	text	utf8_general_ci		Да	NULL		
8	working_regime	varchar(255)	utf8_general_ci		Да	NULL		
9	vk	varchar(255)	utf8_general_ci		Да	NULL		
10	instagram	varchar(255)	utf8_general_ci		Да	NULL		
11	facebook	varchar(255)	utf8_general_ci		Да	NULL		
12	map	text	utf8_general_ci		Да	NULL		

Рис. 1. Таблица «safe»

category – таблица (рис.2) содержит данные о категориях.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	cat_id 	int(11)		UNSIGNED	Нет	Нem		AUTO_INCREMENT
2	category_title	varchar(255)	utf8_general_ci		Нет	Нem		

Рис. 2. Таблица «category»

pars_food – таблица (рис.3) содержит данные о селекторах.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	pars_id 	int(11)			Нет	Нem		AUTO_INCREMENT
2	pars_product	varchar(255)	utf8_general_ci		Нет	Нem		
3	pars_food_img	varchar(255)	utf8_general_ci		Да	NULL		
4	pars_food_title	varchar(255)	utf8_general_ci		Нет	Нem		
5	pars_food_weight	varchar(255)	utf8_general_ci		Да	NULL		
6	pars_food_description	varchar(255)	utf8_general_ci		Да	NULL		
7	pars_food_price	varchar(255)	utf8_general_ci		Нет	Нem		
8	selectors_del	text	utf8_general_ci		Да	NULL		
9	category_url	varchar(255)	utf8_general_ci		Нет	Нem		
10	category_id	int(11)			Нет	Нem		
11	img_url	varchar(255)	utf8_general_ci		Да	NULL		
12	cafe_id	int(11)			Нет	Нem		

Рис. 3. Таблица «pars_food»

food – таблица (рис.4) содержит данные о блюдах и напитках.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	f_id 🗄	int(11)		UNSIGNED	Нет	Нem		AUTO_INCREMENT
2	food_img	varchar(255)	utf8_general_ci		Да	NULL		
3	food_title	varchar(255)	utf8_general_ci		Нет	Нem		
4	weight	varchar(255)	utf8_general_ci		Да	NULL		
5	food_description	text	utf8_general_ci		Да	NULL		
6	price	varchar(255)	utf8_general_ci		Нет	Нem		
7	category_url	varchar(255)	utf8_general_ci		Нет	Нem		
8	category_id 🗄	int(11)		UNSIGNED	Нет	Нem		
9	cafe_id 🗄	int(11)		UNSIGNED	Нет	Нem		
10	img_url	varchar(255)	utf8_general_ci		Да	NULL		

Рис. 4. Таблица «food»

users – таблица (рис.5) содержит данные о администраторе.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id 🗄	int(11)		UNSIGNED	Нет	Нem		AUTO_INCREMENT
2	user	varchar(100)	utf8_general_ci		Нет	Нem		
3	password	varchar(255)	utf8_general_ci		Нет	Нem		
4	auth_key	varchar(255)	utf8_general_ci		Да	NULL		

Рис. 5. Таблица «users»

Таблица food связан с таблицами café и category с типом отношения «один ко многим». Таблицы café и category являются главными, а таблица food – подчиненной. Остальные таблицы являются дополнительными.

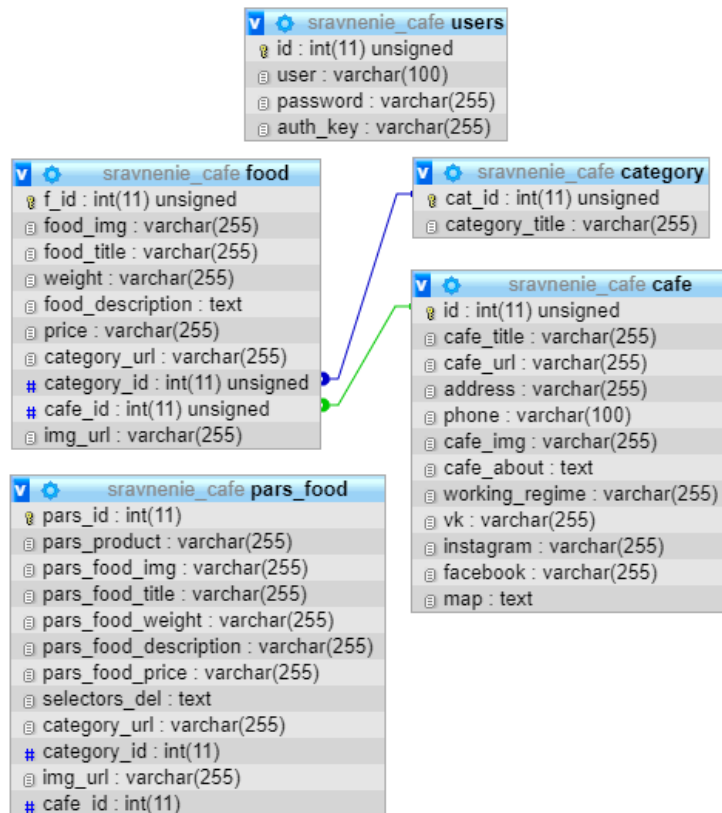


Рис. 6. Схема таблиц

2.1. Структура web-приложения

Пользовательская часть web-приложения

Пользовательская часть web-приложения запускается с файла main.php, которая является главной страницей сайта (рис.7). На данной странице размещены следующие элементы:

- Шапка web-страницы. Шапка состоит из кнопок «Главная» (для возврата на главную страницу), «Кафе» (для перехода на страницу с информацией о кафе города Кызыл) и логотипа «sравnenie.safe» (при нажатий на логотип перекинет на «Главную»).
- Панель фильтрации. С помощью фильтрации пользователь может находить блюда и напитки по названию или по составу, а также сортировать их по категориям, по кафе и по цене.
- Основной контент. В основном контенте выводятся спарсенные (парсить – сбор информации) блюда и напитки. Они выводятся в виде блоков. В верхней части основного контента расположена кнопка «Сравнить». Это кнопка предназначена для сравнения блюд и напитков. Для того чтобы сравнить эти товары, нужно сначала выбрать (нажать на чекбокс «сравнить») их, затем нажать на кнопку сравнения. После нажатия кнопки программа направит на страницу «Сравнение».

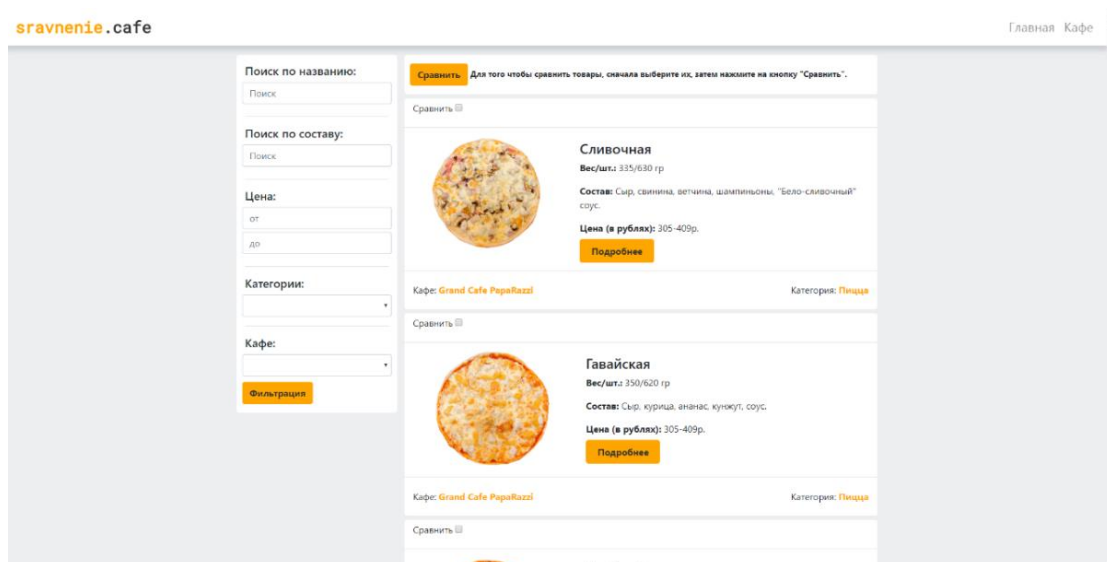


Рис.7. Интерфейс главной страницы сайта

На странице «Сравнение» товары расположены в ряд так, чтобы пользователю было удобно сравнить цены и состав блюда (рис. 8).

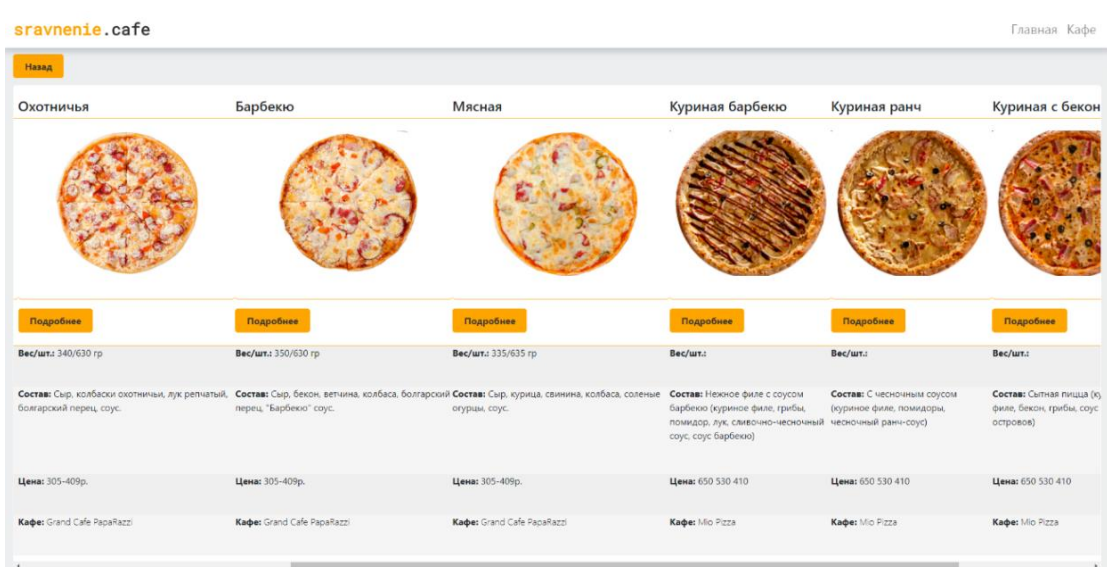


Рис. 8. Интерфейс страницы «Сравнение»

Страница «Кафе», как и в главной странице, состоит из шапки и основного контента (рис. 9). Она содержит блоки. В каждом блоке содержится краткая информация о кафе (изображения, названия, адрес, телефон и сайт) и кнопка «Подробнее». При нажатии на кнопку «Подробнее» программа откроет страницу с подробной информацией о кафе (рис. 10, 11).

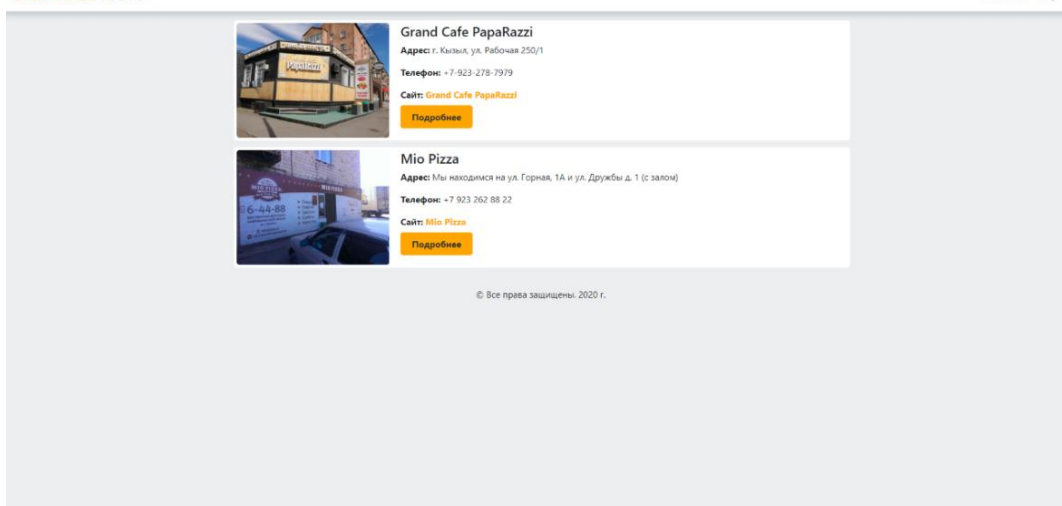


Рис. 9. Страница «Кафе»

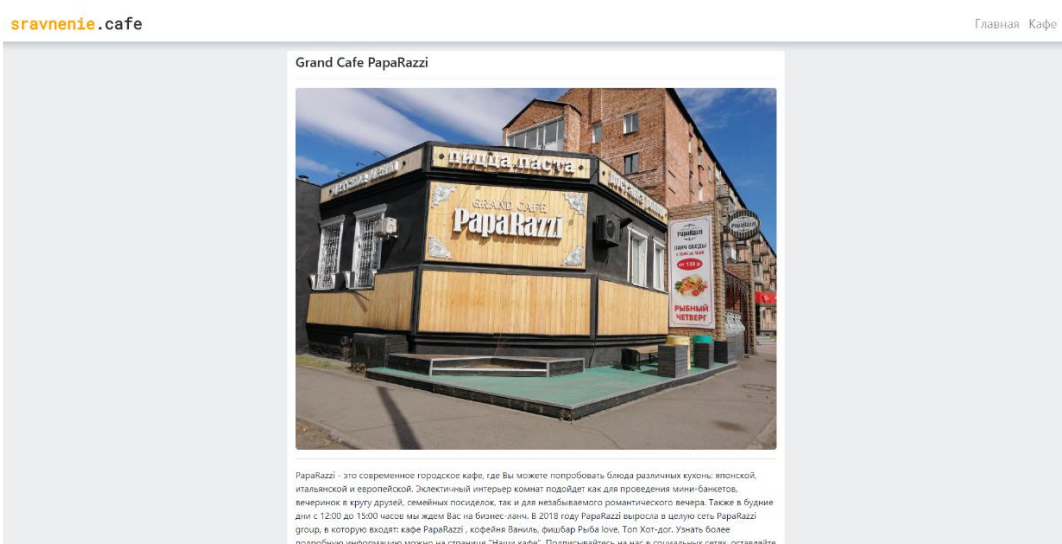


Рис. 11. Подробная информация о кафе

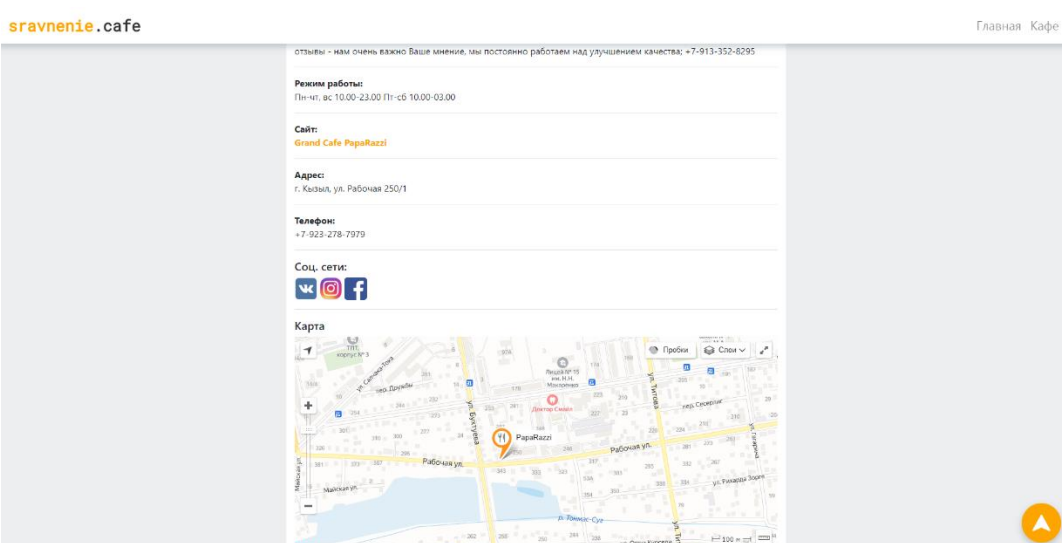
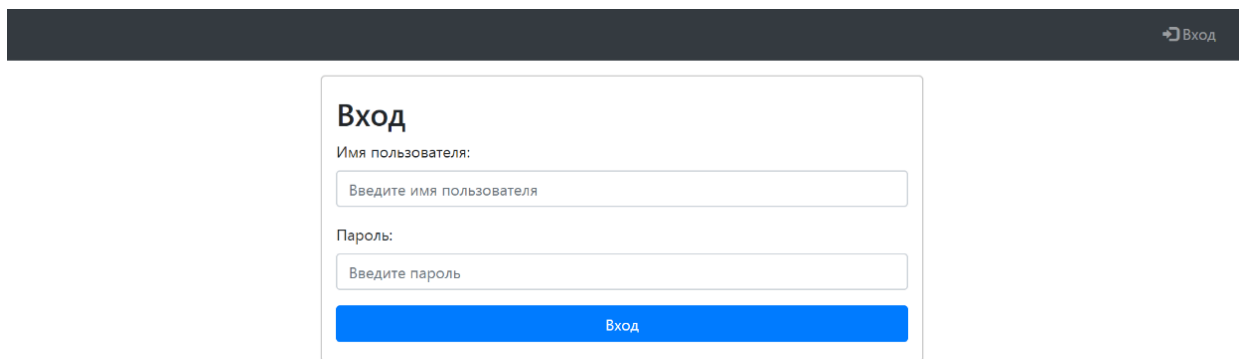


Рис. 12. Подробная информация о кафе

Административная часть web-приложения

Для входа в административную часть нужно на адресной строке браузера ввести следующую строку: адрес сайта/admin, после чего откроется страница «Вход» (рис. 13). На этой странице нужно ввести имя пользователя и пароль. После входа откроется страница администратора сайта. Администраторский панель запускается с страницы «Селекторы блюда» (рис.14). На панели администратора можно осуществлять переход между страницами «Селекторы блюда», «Кафе», «Категории», «Cron». На страницах «Селекторы блюда», «Кафе», «Категории» можно добавлять, удалять, редактировать записи таблиц «pars_food», «cafe», «category». А на странице «Cron» можно посмотреть результат выполнения планировщика (рис. 15).



Вход

Имя пользователя:

Введите имя пользователя

Пароль:

Введите пароль

Вход

Рис. 13. Страница «Вход»

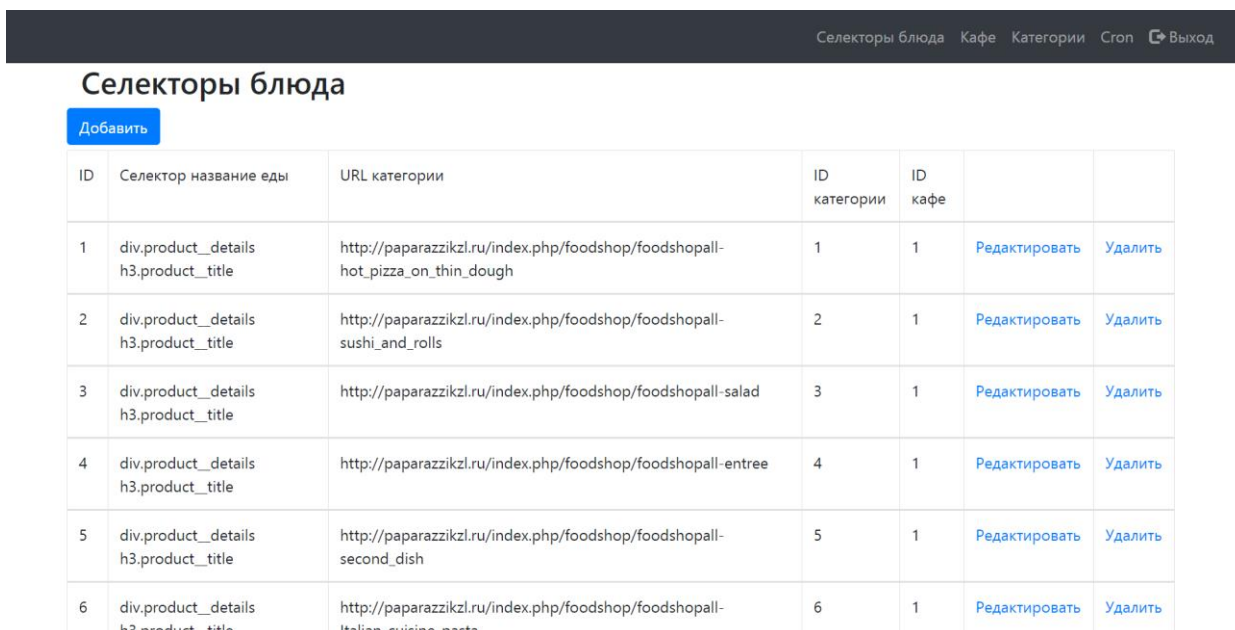


Рис. 14. Страница «Селекторы блюда»

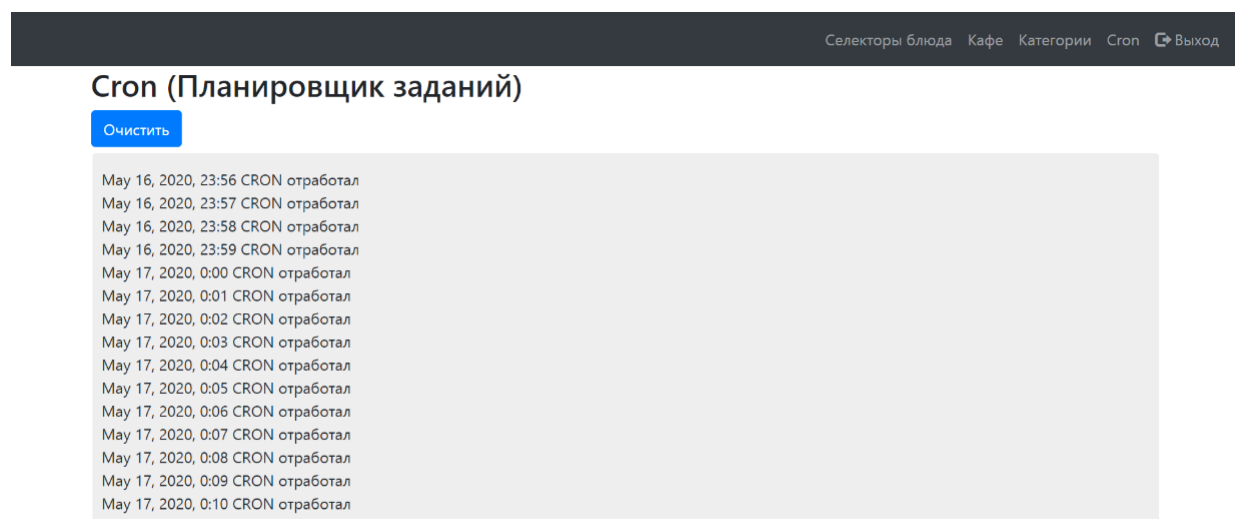


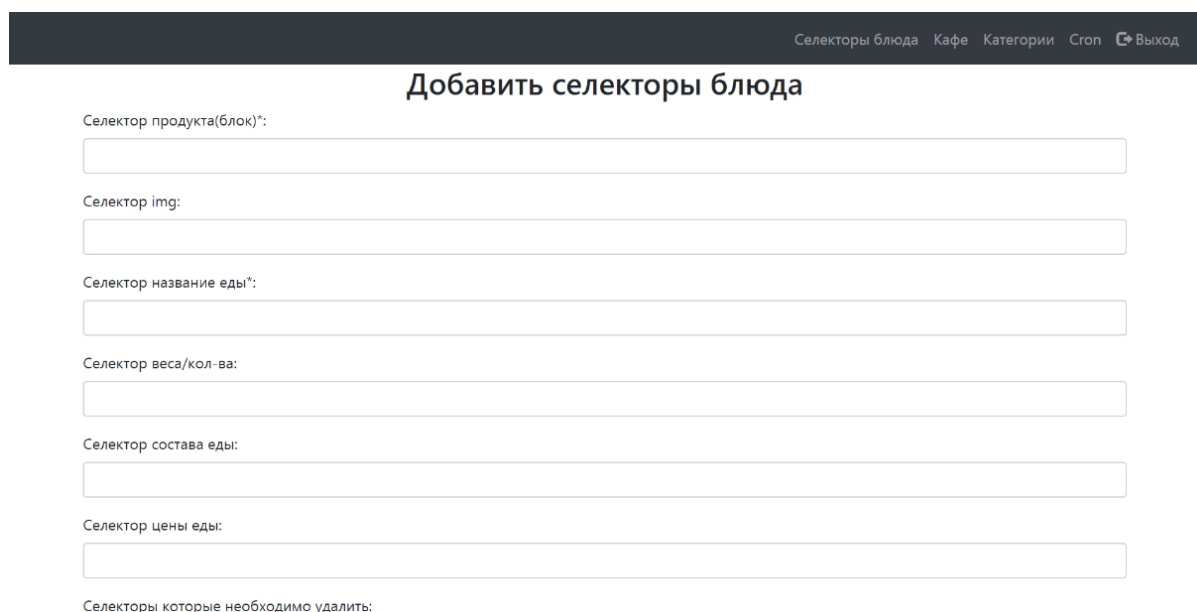
Рис. 15. Страница планировщика заданий

На каждой странице панели администратора находятся следующие элементы:

- Шапка сайта. Шапка состоит из кнопок «Селекторы блюда» (переход на страницу для добавления, редактирования, удаления записей таблицы «pars_food»), «Кафе» (переход на страницу для добавления, редактирования, удаления записей таблицы «safe»), «Категории» (переход на страницу для добавления, редактирования, удаления записей таблицы «category»), «Cron»

(переход на страницу планировщика заданий) и «Выход» (выход из панели администратора).

- Основной контент. На страницах «Селекторы блюда», «Кафе», «Категория» в контенте выводится таблица с записями. Справа от каждой записи таблицы находятся кнопки «Редактировать» и «Удалить». Над таблицей расположена кнопка «Добавить», для добавления записей. Нажав на кнопку администратор попадет на страницу добавления записей. В этой странице находится форма заполнения. Обязательные для заполнения поля отмечены звездочками. А на основном контенте страницы «Сron» выводятся результаты выполнения планировщика заданий (cron). Планировщик заданий запускает файл, в котором написан скрипт для парсинга информации о блюдах и напитках со сайтов кафе города Кызыл.



The screenshot shows a web interface for adding dish selectors. At the top, there is a dark navigation bar with links: 'Селекторы блюда', 'Кафе', 'Категории', 'Сron', and 'Выход'. Below the navigation bar, the main heading is 'Добавить селекторы блюда'. The form consists of several input fields, each with a label: 'Селектор продукта(блок)*:', 'Селектор img:', 'Селектор название еды*:', 'Селектор веса/кол-ва:', 'Селектор состава еды:', and 'Селектор цены еды:'. Each label is followed by a text input field. At the bottom of the form, there is a label 'Селекторы которые необходимо удалить:' followed by an empty input field. The form is styled with a light gray background and white input fields.

Рис. 16. Страница добавления селекторов блюда

Селекторы блюдаКафеКатегорииСнопВыход

Редактирование селекторы блюда с ID = 1

```
string(41) "SELECT * FROM pars_food WHERE pars_id = 1"
```

Селектор продукта(блок)*:

div,product

Селектор img:

div,product_img-holder img,product_img

Селектор название еды*:

div,product_details h3,product_title

Селектор веса/кол-ва:

div,product_details div.anonsn0

Селектор состава еды:

div,product_details div.anonsn

Селектор цены еды:

div.quickview-popup_padding-box h4,foodshop_price

Рис. 17. Страница редактирования селекторов блюда

Селекторы блюдаКафеКатегорииСнопВыход

Удаление селекторы блюда

Удалить запись с ID = 1

Вы действительно хотите удалить эту запись?

Назад к таблице

Удалить

Рис. 18. Страница удаления селекторов блюда

2.2. Использование PHP, phpQuery и MySQL для парсинга информации с сайтов кафе

Для того чтобы получить необходимую информацию с сайтов кафе первым делом, нужно получить html содержимое нужной страницы сайта для парсинга. Для этого было использовано библиотека cURL. Она предназначена для получения и передачи данных через такие протоколы, как HTTP, FTP, HTTPS.

После получение html содержимого страницы сайта парсер начинает собирать данные из указанных веб-ресурсов. Парсер – это программа, сервис или скрипт, который собирает данные с указанных веб-ресурсов, анализирует их и выдает в нужном формате. Для настройки парсера используются различные селекторы контента. Селектор – это правило, по которому можно выделить нужный элемент. Поэтому сначала в администраторской панели в разделе «Селекторы блюда» в форму для ввода вводятся селекторы веб-страницы сайта кафе, в которых содержится необходимая информация. Эти селекторы хранятся в таблице «pars_food».

В планировщике заданий создана задача, которая каждый день запускает файл ParsFood.php. В данном файле хранится скрипт с парсером. В файле ParsFood.php прописан скрипт, алгоритм который работает следующим образом: сначала очищается таблица «food», затем с помощью библиотеки cURL скрипт получает html содержимое из страницы сайта кафе, дальше из таблицы «pars_food» берутся селекторы, и эти селекторы используются в качестве аргументов методов библиотеки phpQuery для парсинга данных. Полученные данные заносятся в таблицу «food» и в дальнейшем они выводятся в пользовательской части Web-приложения для сравнения цен в кафе города Кызыл.

Заключение

В выпускной квалификационной работе решены следующие задачи исследования:

- изучена литература по web-программированию;
- спроектирована и создана база данных для web-приложения;
- подключена библиотека phpQuery к web-приложению для парсинга данных;
- создана панель администратора, позволяющая администратору добавлять, редактировать и удалять записи о кафе и о еде;
- написан скрипт, который сортирует товары (блюда) по цене;
- разработано web-приложение для сравнения цен в кафе города Кызыл.

Разработанное web-приложение для сравнения цен в кафе города Кызыл позволяет: осуществлять навигацию по сайту; просматривать актуальную информацию о товарах на одном сайте, которая будет обновляться ежедневно; осуществлять поиск товаров по её названию, составу, а также сортировать товары по категориям, по цене, по кафе; сравнивать блюда и напитки; владельцам кафе проводить мониторинг ассортиментов и цен конкурентов на одном сайте; узнать информацию о кафе.

Страницы данного сайта разработаны в удобном, красивом интерфейсе, понятном для пользователя. Созданное web-приложение будет использоваться жителями и гостями города Кызыл.

Результаты выпускной квалификационной работы докладывались на ежегодной научно-практической конференции студентов Тувинского государственного университета (Кызыл, 2020).

Список использованной литературы

- 1 Астахова И.В., Мельников В.М., Толстобров А.П., Фертиков В.В. СУБД. Язык SQL в примерах и задачах – ФИЗМАЛИТ, 2007. 168 с.
- 2 Введение в MySQL [Электронный ресурс]. – URL: <https://metanit.com/sql/mysql/1.1.php> (дата обращения: 14.01.2019 г.)
- 3 Примеры использования PDO MySQL [Электронный ресурс]. – URL: <https://snipp.ru/php/manual-pdo> (дата обращения: 14.01.2019 г.)
- 4 Далаа С.М. Основы WEB-программирования: Электр.учеб. пособие. – Кызыл, 2018. (дата обращения: 15.01.2019 г.)
- 5 Учебник «Руководство по PHP» [Электронный ресурс]. – URL: <http://php.net/manual/ru/langref.php> (дата обращения: 01.02.2020).
- 6 Работа с библиотекой phpQuery в PHP [Электронный ресурс]. URL: <http://old.code.mu/books/advanced/php/parsing/rabota-s-bibliotekoj-phpquery-v-php.html> (дата обращения: 12.02.2020 г.).
- 7 Документация phpQuery [Электронный ресурс]. URL: <https://code.google.com/archive/p/phpquery/> (дата обращения: 08.02.2020 г.).

Приложения 1

Администраторская часть сайта

Файл index.php:

```
<?php error_reporting(E_ALL & ~E_NOTICE);
session_start();
if ($_SESSION['admin'] != true) {
    header("Location: ./auth/login.php");
}
else {
    if (isset( $_GET['action'] )) {
        $action = htmlentities($_GET['action']);
    }
    else {
        $action = "";
    }
    switch ( $action ) {
        //cafe
        case 'cafe':
            $results['pageTitle'] = "Кафе";
            require_once("../db.php");
            require_once("../classes/functions.php");
            require_once("../classes/DB.php");
            require_once( "layouts/header_admin.php" );
            require( "cafe/cafe.php" );
            require_once( "layouts/footer_admin.php" );
            break;
        case 'add_cafe':
            $results['pageTitle'] = "Добавление записи в Кафе";
            require_once("../db.php");
            require_once("../classes/functions.php");
            require_once("../classes/DB.php");
            require_once( "layouts/header_admin.php" );
            require( "cafe/add_cafe.php" );
            require_once( "layouts/footer_admin.php" );
            break;
        case 'edit_cafe':
```

```

$results['pageTitle'] = "Редактирование записи в Кафе";
require_once("../db.php");
require_once("../classes/functions.php");
require_once("../classes/DB.php");
require_once( "layouts/header_admin.php" );
require( "cafe/edit_cafe.php" );
require_once( "layouts/footer_admin.php" );
break;
case 'delete_cafe':
    $results['pageTitle'] = "Удаление записи в Кафе";
    require_once("../db.php");
    require_once("../classes/functions.php");
    require_once("../classes/DB.php");
    require_once( "layouts/header_admin.php" );
    require( "cafe/delete_cafe.php" );
    require_once( "layouts/footer_admin.php" );
    break;
//category
case 'category':
    $results['pageTitle'] = "Категории";
    require_once("../db.php");
    require_once("../classes/functions.php");
    require_once("../classes/DB.php");
    require_once( "layouts/header_admin.php" );
    require( "category/category.php" );
    require_once( "layouts/footer_admin.php" );
    break;
case 'add_category':
    $results['pageTitle'] = "Добавление записи в Категории";
    require_once("../db.php");
    require_once("../classes/functions.php");
    require_once("../classes/DB.php");
    require_once( "layouts/header_admin.php" );
    require( "category/add_category.php" );
    require_once( "layouts/footer_admin.php" );
    break;
case 'edit_category':
    $results['pageTitle'] = "Редактирование записи в Категории";

```

```

        require_once("../db.php");
        require_once("../classes/functions.php");
        require_once("../classes/DB.php");
        require_once( "layouts/header_admin.php" );
        require( "category/edit_category.php" );
        require_once( "layouts/footer_admin.php" );
        break;
    case 'delete_category':
        $results['pageTitle'] = "Удаление записи в Категории";
        require_once("../db.php");
        require_once("../classes/functions.php");
        require_once("../classes/DB.php");
        require_once( "layouts/header_admin.php" );
        require( "category/delete_category.php" );
        require_once( "layouts/footer_admin.php" );
        break;
    //food
    case 'food':
        $results['pageTitle'] = "Меню";
        require_once("../db.php");
        require_once("../classes/functions.php");
        require_once("../classes/DB.php");
        require_once( "layouts/header_admin.php" );
        require( "food/food.php" );
        require_once( "layouts/footer_admin.php" );
        break;
    case 'add_food':
        $results['pageTitle'] = "Добавление записи в Меню";
        require_once("../db.php");
        require_once("../classes/functions.php");
        require_once("../classes/DB.php");
        require_once( "layouts/header_admin.php" );
        require( "food/add_food.php" );
        require_once( "layouts/footer_admin.php" );
        break;
    case 'edit_food':
        $results['pageTitle'] = "Редактирование записи в Меню";
        require_once("../db.php");

```

```

        require_once("../classes/functions.php");
        require_once("../classes/DB.php");
        require_once( "layouts/header_admin.php" );
        require( "food/edit_food.php" );
        require_once( "layouts/footer_admin.php" );
        break;
    case 'delete_food':
        $results['pageTitle'] = "Удаление записи в Меню";
        require_once("../db.php");
        require_once("../classes/functions.php");
        require_once("../classes/DB.php");
        require_once( "layouts/header_admin.php" );
        require( "food/delete_food.php" );
        require_once( "layouts/footer_admin.php" );
        break;
    //Планировщик заданий
    case 'cron':
        $results['pageTitle'] = "Cron";
        require_once("../classes/functions.php");
        require_once( "layouts/header_admin.php" );
        require( "cron.php" );
        require_once( "layouts/footer_admin.php" );
        break;
    default:
        $results['pageTitle'] = "Селекторы";
        require_once("../db.php");
        require_once("../classes/functions.php");
        require_once("../classes/DB.php");
        require_once( "layouts/header_admin.php" );
        require( "food/food.php" );
        require_once( "layouts/footer_admin.php" );
    }
} ?>

```

Файл header_admin.php:

```

<?php error_reporting(E_ALL & ~E_NOTICE); ?>
<!DOCTYPE html>
<html lang="ru-Ru">
<head>

```

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, minimum-scale=1.0" />
<link rel="stylesheet" href="../../css/style.css">
<link rel="stylesheet" href="../../css/bootstrap.min.css">
<link rel="stylesheet" href="../../css/font-glyphicons.css">
<title>Админ-панель | <?= htmlspecialchars($results['pageTitle'])?></title>
</head>
<body>
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbar1"
    aria-controls="navbar1" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbar1">
        <?php if(isset($_SESSION['admin']) && $_SESSION['admin']==true) :?>
        <ul class="navbar-nav mr-auto">
            </ul>
            <ul class="nav navbar-nav ml-auto justify-content-end">
                <li class="nav-item"><a class="nav-link" href="index.php?action=food">Селекторы
                блюда</a></li>
                <li class="nav-item"><a class="nav-link" href="index.php?action=cafe">Кафе</a></li>
                <li
                    class="nav-item"><a
                        class="nav-link"
                    href="index.php?action=category">Категории</a></li>
                <li class="nav-item"><a class="nav-link" href="index.php?action=cron">Cron</a></li>
                <li class="nav-item" ><a class="nav-link" href="../../auth/logout.php"><span class="glyphicon
                glyphicon-log-out"></span> Выход</a></li>
            </ul>
            <?php else : ?>
            <ul class="nav navbar-nav ml-auto justify-content-end">
                <li
                    class="nav-item"><a
                        class="nav-link"
                    href="../../auth/login.php"><span
                    class="glyphicon glyphicon-log-in"></span> Вход</a></li>
            </ul>
            <?php endif; ?>
        </div>
    </nav>

    Файл footer_admin.php:
    <script src="../../js/jquery-slim.min.js"></script>
    <script src="../../js/popper.min.js"></script>

```

```

<script src="../../js/bootstrap.min.js"></script>
</body>
</html>

```

Файл login.php:

```

<?php
error_reporting(E_ALL & ~E_NOTICE);
session_start();
if ($_SESSION['admin'] == true)
    header("Location: ../index.php");
else {
    $results['pageTitle'] = "Вход";
    require '../classes/functions.php';
    require '../db.php';
    if(isset($_POST['user']) && isset($_POST['password'])) {
        if (!empty($_POST['user']) && !empty($_POST['password'])) {
            $user = clean_var($_POST['user']);
            $password = clean_var(md5(md5(md5($_POST['password']))));
            $params = array(':user' => $user, ':password' => $password);
            $st = $conn->prepare('SELECT user, password FROM users WHERE user =
:user AND password = :password');
            $st->execute($params);
            $result = $st->fetch(PDO::FETCH_OBJ);
            if (isset($result)) {
                if ($result->user == $user) {
                    $_SESSION['admin'] = true;
                    header("Location: ../index.php");
                }
                else {
                    $errmsg = "Вы ввели неверный логин или пароль.";
                }
            }
        }
    }
    else {
        $errmsg = "Заполните поля Имя пользователя и Пароль.";
    }
}
require '../layouts/header_admin.php';
?>

```

```

<section class="container-fluid">
<div class="login">
    <h2>Вход</h2>
    <?php
        if(isset($errmsg)) {
            echo '<div class="alert alert-
danger"><strong>Ошибка!</strong>'.$errmsg.'</div>';
        }
    ?>
    <form action="" method="post">
        <div class="form-group">
            <label for="user">Имя пользователя:</label>
            <input type="text" class="form-control" id="user" name="user"
placeholder="Введите имя пользователя" value='<?=$_POST['user']; ?>' >
        </div>
        <div class="form-group">
            <label for="password">Пароль:</label>
            <input type="password" class="form-control" id="password"
name="password" placeholder="Введите пароль" >
        </div>
        <button type="submit" class="btn btn-primary">Вход</button>
    </form>
</div>
</section>
<?php require '../layouts/footer_admin.php';?>
<?php } ?>

```

Файл logout.php:

```

<?php error_reporting(E_ALL & ~E_NOTICE);
session_start();
session_destroy(); ?>
<?php include '../layouts/header_admin.php';?>
<section class="container-fluid">
    <div class="login">
        <p>Выход выполнен</p>
        <a href="../auth/login.php">Вернуть в login</a>
        <a href="../index.php">Вернуть к сайту</a>
    </div>
</section>

```

```
<?php include '../layouts/footer_admin.php';?>
```

Файл food.php:

```
<?php
```

```
error_reporting(E_ALL & ~E_NOTICE);
```

```
session_start();
```

```
if ($_SESSION['admin'] != true) :
```

```
    header("Location: ../auth/login.php");
```

```
else :
```

```
    $table = 'pars_food';
```

```
?>
```

```
<section class="container">
```

```
    <h2>Селекторы блюда</h2>
```

```
    <div class="row" >
```

```
    <a class="btn btn-primary" href="index.php?action=add_food" role="button">Добавить</a>
```

```
    <table class="table table-bordered" style="margin: 5px auto;">
```

```
        <thead class="thead-dark">
```

```
            <tr><td>ID</td>
```

```
            <td>Селектор название еды</td>
```

```
            <td>URL категории</td>
```

```
            <td>ID категории</td>
```

```
            <td>ID кафе</td>
```

```
            <td></td>
```

```
            <td></td></tr>
```

```
        </thead>
```

```
    <?php
```

```
        $query = "SELECT * FROM $table";
```

```
        $st = $conn->prepare($query);
```

```
        $st->execute();
```

```
        $rows = $st->fetchAll(PDO::FETCH_ASSOC);
```

```
        foreach ($rows as $row) :?>
```

```
            <tbody>
```

```
                <td><?php echo $row['pars_id']; ?></td>
```

```
                <td><?php echo $row['pars_food_title']; ?></td>
```

```
                <td><?php echo $row['category_url']; ?></td>
```

```
                <td><?php echo $row['category_id']; ?></td>
```

```
                <td><?php echo $row['cafe_id']; ?></td>
```

```
            <td><a href="index.php?action=edit_food&edit_id=<?php echo $row['pars_id']; ?>"
title="Редактировать"><i class="fa fa-pencil-square-o"></i>Редактировать</a></td>
```

```

        <td><a href="index.php?action=delete_food&del_id=<?php echo $row['pars_id']; ?>"
title="Удалить"><i class="fa fa-times"></i>Удалить</a></td>
    </tbody>
</table>
</div>

```

```

</section>

```

```

<?php endif;?>

```

Файл add_food.php:

```

<?php error_reporting(E_ALL & ~E_NOTICE);
session_start();
if ($_SESSION['admin'] != true) :
    header("Location: ../auth/login.php");
else : ?>
<section class="container">
<h2 style="text-align: center;">Добавить селекторы блюда</h2>
<div class="add">
<form action="" method="post" enctype="multipart/form-data">
    <div class="form-group">
        <label for="pars_product">Селектор продукта(блок)*:</label>
        <input type="text" class="form-control" id="pars_product" name="pars_product"
placeholder="" value='<?=$_POST['pars_product']?>' >
    </div>
    <div class="form-group">
        <label for="pars_food_img">Селектор img:</label>
        <input type="text" class="form-control" id="pars_food_img" name="pars_food_img"
placeholder="" value='<?=$_POST['pars_food_img']?>' >
    </div>
    <div class="form-group">
        <label for="pars_food_title">Селектор название еды*</label>
        <input type="text" class="form-control" id="pars_food_title" name="pars_food_title"
placeholder="" value='<?=$_POST['pars_food_title']?>' >
    </div>
    <div class="form-group">
        <label for="pars_food_weight">Селектор веса/кол-ва:</label>
        <input type="text" class="form-control" id="pars_food_weight"
name="pars_food_weight" placeholder="" value='<?=$_POST['pars_food_weight']?>' >
    </div>
    <div class="form-group">

```

```

        <label for="pars_food_description">Селектор состава еды:</label>
        <input      type="name"      class="form-control"      id="pars_food_description"
name="pars_food_description" placeholder="" value='<?=$_POST['pars_food_description']?>' >
    </div>
    <div class="form-group">
        <label for="pars_food_price">Селектор цены еды:</label>
        <input type="text" class="form-control" id="pars_food_price" name="pars_food_price"
placeholder="" value='<?=$_POST['pars_food_price']?>' >
    </div>
    <div class="form-group">
        <label for="selectors_del">Селекторы которые необходимо удалить:</label>
        <textarea      class="form-control"      id="selectors_del"      name="selectors_del"
rows="7"><?=$_POST['selectors_del']?></textarea>
    </div>
    <div class="form-group">
        <label for="category_url">URL категории*:</label>
        <input type="text" class="form-control" id="category_url" name="category_url"
placeholder="" value='<?=$_POST['category_url']?>' >
    </div>
    <div class="form-group">
        <label for="category_id">ID категории*:</label>
        <input type="text" class="form-control" id="category_id" name="category_id"
placeholder="" value='<?=$_POST['category_id']?>' >
    </div>
    <div class="form-group">
        <label for="img_url">URL img:</label>
        <input type="text" class="form-control" id="img_url" name="img_url" placeholder=""
value='<?=$_POST['img_url']?>' >
    </div>
    <div class="form-group">
        <label for="cafe_id">ID кафе:</label>
        <input type="text" class="form-control" id="cafe_id" name="cafe_id" placeholder=""
value='<?=$_POST['cafe_id']?>' >
    </div>
    <button type="submit" class="btn btn-primary">Добавить данные в базу</button></form>
</div></section>
<?php $table = 'pars_food';
$files = "NULL";

```

```

if (!empty($_POST['pars_food_title']) && !empty($_POST['category_id'])) {
    echo DB::add_vals($_POST, $table, $files);
}
else {
    echo "<samp><h4>Вы не ввели обязательные значения! (обязательные значения
обозначены звездочкой *)</h4></samp>";
}
endif;?>

```

Файл edit_food.php:

```

<?php error_reporting(E_ALL & ~E_NOTICE);
session_start();
if ($_SESSION['admin'] != true) :
    header("Location: ../auth/login.php");
else :
    $table = 'pars_food'; ?>
<section class="container">
<h2 style="text-align: center;">Редактирование селекторы блюда с ID = <?php echo
htmlentities($_GET['edit_id']);?></h2>
<div class="add">
<?php
if (isset($_GET['edit_id']) && (!empty($_GET['edit_id']))) {
    $id_edit = htmlentities($_GET['edit_id']);
    $query = "SELECT * FROM $table WHERE pars_id = $id_edit";
    $st = $conn->prepare($query);
    $st->execute();
    $rows = $st->fetchAll(PDO::FETCH_ASSOC);
    foreach ($rows as $row) {}
}??>
<form action="" method="post" enctype="multipart/form-data">
    <div class="form-group">
        <label for="pars_product">Селектор продукта(блок)*:</label>
        <input type="text" class="form-control" id="pars_product" name="pars_product"
placeholder="" value='<?=$row['pars_product'];?>' >
    </div>
    <div class="form-group">
        <label for="pars_food_img">Селектор img:</label>
        <input type="text" class="form-control" id="pars_food_img" name="pars_food_img"
placeholder="" value='<?=$row['pars_food_img'];?>' >

```

```

</div>
<div class="form-group">
    <label for="pars_food_title">Селектор название еды*:</label>
    <input type="text" class="form-control" id="pars_food_title" name="pars_food_title"
placeholder="" value='<?=$row['pars_food_title'];?>' >
</div>
<div class="form-group">
    <label for="pars_food_weight">Селектор веса/кол-ва:</label>
    <input type="text" class="form-control" id="pars_food_weight"
name="pars_food_weight" placeholder="" value='<?=$row['pars_food_weight'];?>' >
</div>
<div class="form-group">
    <label for="pars_food_description">Селектор состава еды:</label>
    <input type="name" class="form-control" id="pars_food_description"
name="pars_food_description" placeholder="" value='<?=$row['pars_food_description'];?>' >
</div>
<div class="form-group">
    <label for="pars_food_price">Селектор цены еды:</label>
    <input type="text" class="form-control" id="pars_food_price" name="pars_food_price"
placeholder="" value='<?=$row['pars_food_price'];?>' >
</div>
<div class="form-group">
    <label for="selectors_del">Селекторы которые необходимо удалить:</label>
    <textarea class="form-control" id="selectors_del" name="selectors_del"
rows="7"><?=$row['selectors_del'];?></textarea>
</div>
<div class="form-group">
    <label for="category_url">URL категории*:</label>
    <input type="text" class="form-control" id="category_url" name="category_url"
placeholder="" value='<?=$row['category_url'];?>' >
</div>
<div class="form-group">
    <label for="category_id">ID категории*:</label>
    <input type="text" class="form-control" id="category_id" name="category_id"
placeholder="" value='<?=$row['category_id'];?>' >
</div>
<div class="form-group">
    <label for="img_url">URL img:</label>

```

```

        <input type="text" class="form-control" id="img_url" name="img_url" placeholder=""
value='<?=$row['img_url'];?>' >
    </div>
    <div class="form-group">
        <label for="cafe_id">ID cafe*:</label>
        <input type="text" class="form-control" id="cafe_id" name="cafe_id" placeholder=""
value='<?=$row['cafe_id'];?>' >
    </div>
    <button type="submit" class="btn btn-primary">Редактировать</button></form></div>
</section>
<?php $files = "NULL";
    echo DB::update_vals($_POST, $table, $files);
endif; ?>

```

Файл delete_food.php:

```

<?php error_reporting(E_ALL & ~E_NOTICE);
session_start();
if ($_SESSION['admin'] != true) :
    header("Location: ../auth/login.php");
else : ?>
<section class="container">
    <h2 style="text-align: center;">Удаление селекторы блюда</h2>
    <h4>Удалить запись с ID = <?php echo $_GET['del_id'];?></h4>
    <p>Вы действительно хотите удалить эту запись?</p>
    <div><a class="btn btn-primary" href="index.php?action=food" role="button">Назад к
таблице</a></div>
    <form method="post"><input type="submit" name="submit" value="Удалить" /></form>
</section>
<?php $table = 'pars_food';
    echo DB::delete($_GET['del_id'], $table);
endif;?>

```

Файл cron.php:

```

<section class="container">
<?php echo "<h2>Cron (Планировщик заданий)</h2>";
    echo "<form method='post'>
    <input type='submit' name='clear' class='btn btn-primary' value='Очистить'>
    </form>";
    $text = file_get_contents("../classes/test.txt");
    if (!empty($text)) {

```

```
        echo "<div style='background: #eee; border: 1px solid #fff; border-radius: 5px; margin:
5px 0; padding: 15px 10px'>".$text."</div><br>";
    }
    else echo "<p>Файл пуст!</p>";
    if(isset($_POST['clear']))
    {
        $f = fopen('../classes/test.txt','w');
        ftruncate($f, 0);
        fclose($f);
    } ?>
</section>
```

Приложения 2

Пользовательская часть сайта

Файл index.php:

```
<?php
error_reporting(0);
if (isset( $_GET['action'] )) {
    $action = htmlentities($_GET['action']);
}
else {
    $action = "";
}
switch ( $action ) {
//food
case 'main':
    $results['meta_d'] = 'Главная страница. Сравнение цен в кафе города Кызыл';
    $results['meta_k'] = 'Главная, страница, сравнение цен в кафе города Кызыл, кафе,
сравнение цен, Кызыл, Республика Тыва';
    $results['pageTitle'] = "Главная";
    require("db.php");
    require("classes/functions.php");
    require("classes/DB.php");
    require("classes/Filters.php");
    require( "layouts/header.php" );
    require("site/main.php");
    require( "layouts/footer.php" );
    break;
//sravnenie
case 'sravfood':
    $results['pageTitle'] = "Сравнение";
    require("db.php");
    require("classes/functions.php");
    require("classes/DB.php");
    require("classes/Filters.php");
    require( "layouts/header.php" );
    require("site/sravfood.php");
    require( "layouts/footer.php" );
```

```

        break;

//cafe
case 'cafe':
    $results['meta_d'] = 'Кафе. Сравнение цен в кафе города Кызыл';
    $results['meta_k'] = 'Кафе, сравнение цен в кафе города Кызыл, кафе,
сравнение цен, Кызыл, Республика Тыва';
    $results['pageTitle'] = "Кафе";
    require("db.php");
    require("classes/functions.php");
    require("classes/DB.php");
    require("classes/Filters.php");
    require( "layouts/header.php" );
    require("site/cafe.php");
    require( "layouts/footer.php" );
    break;

//cafe
case 'cafe_content':
    $results['pageTitle'] = "Кафе";
    require("db.php");
    require("classes/functions.php");
    require("classes/DB.php");
    require("classes/Filters.php");
    require( "layouts/header.php" );
    require("site/cafe_content.php");
    require( "layouts/footer.php" );
    break;

default:
    $results['meta_d'] = 'Главная страница. Сравнение цен в кафе города Кызыл';
    $results['meta_k'] = 'Главная, страница, сравнение цен в кафе города Кызыл, кафе,
сравнение цен, Кызыл, Республика Тыва';
    $results['pageTitle'] = "Главная";
    require("db.php");
    require("classes/functions.php");
    require("classes/DB.php");
    require("classes/Filters.php");
    require( "layouts/header.php" );
    require( "site/main.php" );
    require( "layouts/footer.php" );

```

```
}  
?>
```

Файл main.php:

```
<?php error_reporting(E_ALL & ~E_NOTICE);  
$table = 'food';  
$table2 = 'category';  
$table3 = 'cafe'; ?>  
<section class="snav container">  
<!-- Фильтрация -->  
<div class="filters">  
<form action="" method="post" enctype="multipart/form-data">  
    <h5>Поиск по названию:</h5>  
    <div class="form-row align-items-center">  
        <div class="filter-field">  
            <input class="form-control" name="searchTitle" type="search"  
value="<?=$_POST['searchTitle']?>" placeholder="Поиск">  
        </div>  
    </div><hr>  
    <h5>Поиск по составу:</h5>  
    <div class="form-row align-items-center">  
        <div class="filter-field">  
            <input class="form-control" name="searchDescrip" type="search" value="<?php  
echo $_POST['searchDescrip']?>" placeholder="Поиск">  
        </div>  
    </div><hr>  
    <h5>Цена:</h5>  
    <div class="form-row align-items-center">  
        <div class="filter-field">  
            <input class="form-control" name="fromPrice" type="search" value="<?php  
echo $_POST['fromPrice']?>" placeholder="от">  
            <input class="form-control" name="toPrice" type="search" value="<?php echo  
$_POST['toPrice']?>" placeholder="до">  
        </div>  
    </div><hr>  
    <h5>Категории:</h5>  
    <div class="form-row align-items-center">  
    <select class="form-control" name="category" id="exampleFormControlSelect1">  
        <option ></option>
```

```

        <?php $rows = DB::simple_select($table2);
            foreach ($rows as $row) : ?>
                <option value="<?php echo $row['cat_id'];?>"><?php echo
$row['category_title'];?></option>
            <?php endforeach; ?>
        </select>
    </div><hr>
    <h5>Кафе:</h5>
    <div class="form-row align-items-center">
        <select class="form-control" name="cafe" id="exampleFormControlSelect1">
            <option ></option>
            <?php $rows = DB::simple_select($table3);
                foreach ($rows as $row) : ?>
                    <option value="<?php echo $row['id'];?>"><?php echo $row['cafe_title'];?></option>
                <?php endforeach;?>
            </select><br>
            <button class="btn but-cafe" style="font-weight: bolder; margin-top: 10px;" name="filter"
type="submit">Фильтрация</button>
        </div>
    </form>
</div>
<!-- Контент -->
<div class="content">
    <form action="index.php?action=sravfood" method="post">
    <div class="food" style="padding: 15px 10px">
        <button class="btn but-cafe" style="font-weight: bolder;" name="submit"
type="submit">Сравнить</button>
        <span style="font-size: 14px"><strong>Для того чтобы сравнить товары, сначала выберите
их, затем нажмите на кнопку "Сравнить".</strong></span>
    </div>
    <?php if (isset($_POST['filter'])) {
        $query = Filters::getMainQuery($table, $table2, $table3);
        $where = "";
        if ($_POST["searchTitle"]) {
            $where = Filters::filter($where, "food_title LIKE '%" .
htmlspecialchars($_POST["searchTitle"])) . "%'";
        }
        if ($_POST["searchDescrip"]) {

```

```

        $where = Filters::filter($where, "food_description LIKE '%" .
htmlspecialchars($_POST["searchDescrip"])) . "%";
    }
    if ($_POST["fromPrice"]) {
        $where = Filters::filter($where, "price >= '" . htmlspecialchars($_POST["fromPrice"])) .
        """;
    }
    if ($_POST["toPrice"]) {
        $where = Filters::filter($where, "price <= '" . htmlspecialchars($_POST["toPrice"])) . """;
    }
    if ($_POST["category"]) {
        $where = Filters::filter($where, "category_id = '" . htmlspecialchars($_POST['category']))
        . """;
    }
    if ($_POST["cafe"]) {
        $where = Filters::filter($where, "id = '" . htmlspecialchars($_POST['cafe'])) . """;
    }
    $second = "$table.category_id=$table2.cat_id AND $table.cafe_id=$table3.id";
    if ($where) {
        $query .= " WHERE $where AND ". $second;
    }
    $st = $conn->prepare($query);
    $st->execute();
    $rows = $st->fetchAll(PDO::FETCH_ASSOC);
    if (!empty($rows)) {
        foreach ($rows as $row) {
            require("content_food.php");
        }
    }
    else { echo '<div class="food"><div class="food-content"><samp><h4>По вашему запросу
ничего не найдено</h4></samp></div></div>'; }
}
else {
    $rows = Filters::food_select($table,$table2,$table3);
    foreach ($rows as $row) { require("content_food.php");}
} ?>
</form>
</div>

```

```
<!-- Кнопка вверх -->
```

```
<div class="scrollup"></div>
```

```
</section>
```

Файл content_food.php:

```
<div class="food">
```

```
    <div class="check">
```

```
        <label class="form-check-label">Сравнить</label>
```

```
        <input type="checkbox" name="srav[]" value="<?php echo $row['f_id'];?>">
```

```
</div><hr>
```

```
<div class="food-block">
```

```
    <div class="food-img">
```

```
        <?php echo "<img src='".$row['img_url'].$row['food_img']."'>";?>
```

```
</div>
```

```
    <div class="food-content">
```

```
        <?php echo "<h4>".$row['food_title']."</h4>";?>
```

```
        <?php echo "<p><strong>Вес/шт.: </strong>".$row['weight']."</p>";?>
```

```
        <?php echo "<p><strong>Состав: </strong>".$row['food_description']."</p>";?>
```

```
        <?php echo "<p><strong>Цена (в рублях): </strong>".$row['price']."</p>";?>
```

```
        <?php echo "<a href='".$row['category_url']."' class='but-cafe' role='button' aria-pressed='true'>Подробнее</a>";?>
```

```
    </div>
```

```
</div><hr>
```

```
<div class="food-footer">
```

```
    <?php echo "<p>Кафе: <a href='".$row['cafe_url']."'>".$row['cafe_title']."</a></p>";?>
```

```
    <?php echo "<p>Категория: <a href='".$row['category_url']."'>".$row['category_title']."</a></p>";?>
```

```
</div>
```

```
</div>
```

Файл sravfood.php:

```
<?php error_reporting(E_ALL & ~E_NOTICE);
```

```
$table = 'food'; ?>
```

```
<section class="container-fluid" >
```

```
<?php echo "<div style='margin: 20px 0px'><a href='index.php?action=main' class='but-cafe' role='button' aria-pressed='true'>Назад</a></div>";
```

```
$srav = array();
```

```
echo "<div class='sravfood'>";
```

```
if (isset($_POST) && !empty($_POST['srav'])) {
```

```

$srav = $_POST['srav'];
$i=0;
foreach ($srav as $id_srav) {
    $query = "SELECT * FROM $table, cafe WHERE :id_srav = f_id and cafe_id=id";
    $params = array(':id_srav' => $id_srav);
    $st = $conn->prepare($query);
    $st->execute($params);
    $rows = $st->fetchAll(PDO::FETCH_ASSOC);
    echo '<div class="sravfood_block">';
    foreach ($rows as $row) : ?>
        <div class="sravfood_block_title">
            <?php echo "<h4>".$row['food_title']."</h4>";?>
        </div>
        <div class="sravfood_block_img">
            <?php echo "<img src='".$row['img_url'].$row['food_img']."'>";?>
        </div>
        <div class="sravfood_block_btn">
            <?php echo "<a href='".$row['category_url']."' class='btn-srav'
>Подробнее</a>";?>
        </div>
        <div class="sravfood_block_weight">
            <?php echo "<p><strong>Вес/шт.: </strong>".$row['weight']."</p>";?>
        </div>
        <div class="sravfood_block_desc">
            <?php echo "<p><strong>Состав: </strong>".$row['food_description']."</p>";?>
        </div>
        <div class="sravfood_block_price">
            <?php echo "<p><strong>Цена: </strong>".$row['price']."</p>";?>
        </div>
        <div class="sravfood_block_cafe">
            <?php echo "<p><strong>Кафе: </strong><a class='links' style='color: #000;
font-weight: 100; href='".$row['cafe_url']."'>".$row['cafe_title']."</a></p>";?>
        </div>
    <?php endforeach;
    echo '</div>';
    $i++;
}
}

```

```

else {
    echo "<samp><h4>Вы ничего не выбрали для сравнения!</h4></samp>";
}
echo "</div>";
?>
</section>

```

Файл **cafe.php**:

```

<?php error_reporting(E_ALL & ~E_NOTICE);
$stable = 'cafe'; ?>
<section class="container" style="margin-top: 15px;">
<?php $rows = DB::simple_select($stable);
    foreach ($rows as $row) : ?>
        <div class="cafe">
            <div class="cafe-block">
                <div class="cafe-img"><?php echo "<img src='".$row['cafe_img']."'>";?></div>
                <div class="cafe-content">
                    <?php echo "<h4>".$row['cafe_title']."</h4>";?>
                    <?php echo "<p><strong>Адрес: </strong>".$row['address']."</p>";?>
                    <?php echo "<p><strong>Телефон: </strong>".$row['phone']."</p>";?>
                    <?php echo "<p><strong>Сайт: </strong><a class='links'
href='".$row['cafe_url']."'>".$row['cafe_title']."</a></p>";?>
                    <?php echo "<a href='index.php?action=cafe_content&cafe_con_id='".$row['id']."'
class='but-cafe' role='button' aria-pressed='true'>Подробнее</a>";?>
                </div>
            </div>
        </div>
    <?php endforeach; ?>
<!-- Кнопка вверх -->
<div class="scrollup"></div>
</section>

```

Файл **café_content.php**:

```

<?php error_reporting(E_ALL & ~E_NOTICE);
$stable = 'cafe'; ?>
<section class="container">
<?php $rows = DB::simple_select($stable);
    foreach ($rows as $row) {
        if ($row['id']==$_GET['cafe_con_id']) { ?>
            <div class="cafe-container">

```

```

<div class="cafe-content">
    <?php echo "<h4>".$row['cafe_title']."</h4>";?><hr>
    <?php echo "<img src='".$row['cafe_img']."'>";?><hr>
    <?php echo "<p>".$row['cafe_about']."</p>";?><hr>
    <?php echo "<b>Режим работы:
</b><p>".$row['working_regime']."</p>";?><hr>
    <?php echo "<b>Сайт: </b><p><a class='links'
href='".$row['cafe_url']."'>".$row['cafe_title']."</a></p>";?><hr>
    <?php echo "<b>Адрес: </b><p>".$row['address']."</p>";?><hr>
    <?php echo "<b>Телефон: </b><p>".$row['phone']."</p>";?><hr>
    <h5>Соц. сети: </h5>
    <?php echo "<a href='".$row['vk']."' class='icon-socnet'><img
src='../img/vk.png' alt='vk'></a>";?>
    <?php echo "<a href='".$row['instagram']."' class='icon-socnet'><img
src='../img/instagram.png' alt='instagram'></a>";?>
    <?php echo "<a href='".$row['facebook']."' class='icon-socnet'><img
src='../img/facebook.png' alt='facebook' ></a>";?><hr>
    <h5>Карта</h5>
    <?php echo "<p>".$row['map']."</p>";?>
</div>
</div>
<?php }
} ?>
<!-- Кнопка вверх -->
<div class="scrollup"></div>
</section>

```

Приложения 3

Классы, функции.

Файл db.php:

```
<?php $dns = "mysql:host=127.0.0.1;dbname=sravnenie_cafe;charset=utf8";
$user="root";
$password="";
try {
    $conn = new PDO( $dns, $user, $password, array( PDO::
MYSQL_ATTR_INIT_COMMAND => 'SET NAMES utf8' ) );
}
catch(PDOException $e) {
    echo "К сожалению, произошла ошибка. Пожалуйста, попробуйте позже.";
    error_log( $exception->getMessage() );
} ?>
```

Файл DB.php:

```
<?php class DB {
    //обычная выборка
    public static function simple_select($table) {
        require 'db.php';
        $query = "SELECT * FROM $table";
        $st = $conn->prepare($query);
        $st->execute();
        $rows = $st->fetchAll(PDO::FETCH_ASSOC);
        return $rows;
    }
    //метод для получения столбцов таблиц
    public static function getColumns($table) {
        require "../db.php";
        $query = "DESCRIBE $table";
        $st = $conn->prepare($query);
        debug($query);
        $st->execute();
        $row = $st->fetchAll(PDO::FETCH_COLUMN);
        return $row;
    }
    public static function add_vals($post, $table, $files) {
```

```

if (isset($_POST) && !empty($_POST)) {
    $i=0;
    $j=0;
    foreach ($_POST as $key => $value) {
        if (!empty($value)) {
            $keys[$j] = $key;
            $val[$i] = $value;
        }
        $i++;
        $j++;
    }
    if ($table=="cafe") {
        $imgKey = 'cafe_img';
        $imgVal = img_add($_FILES);
        if (!empty($imgVal)) {
            array_push($keys, $imgKey); //Добавляет один или несколько
элементов в конец массива (1-й параметр - массив; 2-й и до N параметр - элемент (элементы))
            array_push($val, $imgVal);
        }
    }
    self::insert($keys,$val,$table);
}

}

public static function update_vals($post, $table, $files) {
    if(isset($_GET['edit_id']) && !(empty($_POST))) {
        $id = htmlentities($_GET['edit_id']);
        $i=0;
        $j=0;
        foreach ($_POST as $key => $value) {
            if (!empty($value)) {
                $keys[$j] = $key;
                $val[$i] = $value;
            }
            $i++;
            $j++;
        }
        if ($table=="cafe") {
            $imgKey = 'cafe_img';

```

```

        $imgVal = img_add($_FILES);
        if (!empty($imgVal)) {
            array_push($keys, $imgKey);
            array_push($val, $imgVal);
        }
    }
    self::update($keys,$val,$id,$table);
}
}

//метод для добавления записей таблиц
public static function insert($keys,$val,$table) {
    require "../db.php";
    $keys = array_values($keys); // получаем индексный массив
    $val = array_values($val); // получаем индексный массив
    $query = "INSERT INTO $table (";
    $j=0;
    for ($i=0; $i < count($keys); $i++) {
        if (!empty($keys[$i])) {
            if ($i==0) {
                $z = $z.$keys[$i];
            }
            else {
                $z = $z.", ".$keys[$i];
            }
        }
    }
    $query = $query.$z.") VALUES (";
    $i=0;
    for ($i=0; $i < count($keys); $i++) {
        if (!empty($keys[$i])) {
            if ($i==0) {
                if (empty($val[$i])) {
                    $q = $q." ".$keys[$i];
                }
                else {
                    $q = $q." ".$keys[$i];
                }
            }
        }
    }
}

```

```

        else {
            if (empty($val[$i])) {
                $q = $q.", " . ":" . $keys[$i];
            }
            else {
                $q = $q.", " . ":" . $keys[$i];
            }
        }
        $params[$i] = array(':' . $keys[$i] => $val[$i]);
    }
}
$q = $q . ")";
$query = $query . $q;
$params = call_user_func_array('array_merge', $params); //функция, которая объединяет
многомерный массив в одномерный
debug($params);
$stmt = $conn->prepare($query);
if ($stmt->execute($params)) {
    echo "<samp><h4>Запись добавлена!</h4></samp>";
}
else {
    echo "<samp><h4>Запись не добавлена!</h4></samp>";
}
}
//метод для обновления записей таблиц
public static function update($keys,$val,$id,$table) {
    require "../db.php";
    echo $id . "<br>";
    $k=0;
    $m=0;
    $keys = array_values($keys);
    $val = array_values($val);
    switch ($table) {
        case 'pars_food':
            $id_t = 'pars_id';
            break;
        case 'cafe':
            $id_t = 'id';

```

```

        break;
    case 'category':
        $id_t = 'cat_id';
        break;
    default:
        $id_t = 'id';
        break;
}
$query = "UPDATE $table SET ";
for ($i=0; $i<count($keys); $i++) {
    if (!empty($keys[$i])) {
        $m++;
        if ($m>1) {
            $q = $q.", ".$keys[$i];
            $q = $q." = ".$keys[$i];
        }
        else {
            $q = $q.$keys[$i];
            $q = $q." = ".$keys[$i];
        }
        $params[$i] = array('.'.$keys[$i] => $val[$i]);
    }
}
array_push($params, array('id' => $id));
$q = $q." WHERE $id_t = :id";
$query = $query.$q;
$params = call_user_func_array('array_merge', $params);
$stmt = $conn->prepare($query);
if ($stmt->execute($params)) {
    echo "<samp><h4>Запись изменена!</h4></samp>";
}
else {
    echo "<samp><h4>Запись не изменена!";
}
}

//метод для удаления записей таблиц
public static function delete($get, $table) {
    require "../db.php";

```

```

if (isset($_POST['submit'])) {
    if (isset($_GET) && !empty($_GET)) {
        //выбранный id
        $id = htmlentities($_GET['del_id']);
        //определяется к какой табл относиться id
        switch ($table) {
            case 'pars_food':
                $id_t = 'pars_id';
                break;
            case 'cafe':
                $id_t = 'id';
                break;
            case 'category':
                $id_t = 'cat_id';
                break;
            default:
                $id_t = 'id';
                break;
        }
        $query = "DELETE FROM $table WHERE $id_t = :id";
        $params = array(':id' => $id);
        $st = $conn->prepare($query);
        if ($st->execute($params)) {
            echo "<samp><h4>Запись удалена!</h4></samp>";
        }
        else {
            echo "<samp><h4>Запись не удалена!</h4></samp>";
        }
    }
}
} ?>

```

Файл Filters.php:

```

<?php class Filters {
    //filters
    public static function getMainQuery($table, $table2, $table3) {
        return "SELECT * FROM $table, $table2, $table3";
    }
}

```

```

public static function filter($where, $add, $and = true) {
    if ($where) {
        if ($and) $where .= " AND $add";
        else $where .= " OR $add";
    } else $where = $add;
    return $where;
}

//ВЫВОД ДАННЫХ
public static function food_select($table, $table2, $table3) {
    require 'db.php';
    $query = "SELECT * FROM $table, $table2, $table3 WHERE
    $table.category_id=$table2.cat_id and $table.cafe_id=$table3.id";
    $st = $conn->prepare($query);
    $st->execute();
    $rows = $st->fetchAll(PDO::FETCH_ASSOC);
    return $rows;
}
} ?>

```

Файл functions.php:

```

<?php function clean_var($value) {
    $value = trim($value); //удаление пробелов
    $value = stripslashes($value); // удаление экранированных символов
    $value = strip_tags($value); // удаление html и php тегов
    $value = htmlspecialchars($value); // преобразует специальные символы в html-сущности
    $value = htmlentities($value);
    return $value;
}

//функция для вывода информации о переменной (массиве)
function debug($val) {
    echo "<pre>"; var_dump($val); echo "</pre>";
}

//функция для загрузки изображения
function img_add($files) {
    if (!empty($_FILES['file']['tmp_name'])) {
        if (empty($_FILES['file']['error'])) {
            if ($_FILES['file']['type'] == 'image/jpeg' || $_FILES['file']['type'] == 'image/jpg' ||
            $_FILES['file']['type'] == 'image/png') {
                if ($_FILES['file']['size'] <= 2 * 1024 * 1024) {

```

```

        $uploadname=$_FILES['file']['name'];//записываем имя файла
        $dir = "../img/"; // Например /var/www/localhost/public
        if (!file_exists($dir)) { mkdir($dir, 0777, true);}
        $uploadpath=$dir.$uploadname; //указываем куда грузить файл
        if (!move_uploaded_file($_FILES['file']['tmp_name'], $uploadpath)) {
echo 'Ошибка загрузки<br>'; }
        else { return $uploadpath; }
    }
    else { echo "Размер файла превышает 2 мегабайта<br>";}
}
else {echo "Загружайте только jpg-, png-файлы!<br>";}
}
}
else { echo "Вы не загрузили файл!<br>";}
} ?>

```

Файл Parsing.php:

```

<?php
class Parsing {
    public static function get_page_by_url($url){
        $page = self::curl_get($url);
        if (!$page){
            return NULL;
        }
        if (preg_match('# content="text/html; charset=(.+)#"#isU', $page, $matches)) {
            $charset = $matches[1];
        }
        if (isset($charset) && strtolower($charset) != 'utf-8'){
            $page = iconv($charset, 'UTF-8//IGNORE', $page);
            $page = preg_replace('#content="text/html; charset=.'.$charset.'"#isU',
'content="text/html; charset=utf-8"', $page);
        }
        return $page;
    }
    public static function curl_get($url) {
        $use_proxy = 0;
        $proxy = '127.0.0.1:5555';
        $ch = curl_init($url);
    }
}

```

```

curl_setopt($ch, CURLOPT_USERAGENT, 'IE20');
curl_setopt($ch, CURLOPT_HEADER, 0);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, '1');
curl_setopt($ch, CURLOPT_FOLLOWLOCATION, true);
curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, FALSE);
curl_setopt($ch, CURLOPT_SSL_VERIFYHOST, 2);
if ($use_proxy){
    curl_setopt($ch, CURLOPT_PROXYTYPE, CURLPROXY_SOCKS5);
    curl_setopt($ch, CURLOPT_PROXY, $proxy);
}
$html = curl_exec($ch);
curl_close($ch);
return $html;
}
}
?>

```

Приложения 4

Файл ParsFood.php

```
<?php
require_once("../db.php");
require_once("../classes/DB.php");
require_once("../classes/functions.php");
require_once("../classes/phpQuery.php");
require_once("../classes/Parsing.php");
//очистка таблицы food
$st2 = $conn->prepare('TRUNCATE TABLE food');
$st2->execute();
// строка, которую будем записывать
date_default_timezone_set('Asia/Krasnoyarsk');
$text = date("F j, Y, G:i")." CRON отработал\n<br>";
//Открываем файл
$fp = fopen("test.txt", "a+");
// записываем в файл текст
fwrite($fp, $text);
// закрываем
fclose($fp);
//получаем данные (селекторы) из таблицы pars_food
DB::simple_select('pars_food');
$vals = array();
foreach ($rows as $row) {
    $pars_product[] = $row['pars_product'];
    $pars_food_img[] = $row['pars_food_img'];
    $pars_food_title[] = $row['pars_food_title'];
    $pars_food_weight[] = $row['pars_food_weight'];
    $pars_food_description[] = $row['pars_food_description'];
    $pars_food_price[] = $row['pars_food_price'];
    $selectors_del[] = $row['selectors_del'];
    $category_url[] = $row['category_url'];
    $category_id[] = $row['category_id'];
    $img_url[] = $row['img_url'];
    $cafe_id[] = $row['cafe_id'];
}
```

```

$pars_product = array_values($pars_product);//возвращает индексный массив
    $pars_food_img = array_values($pars_food_img);
    $pars_food_title = array_values($pars_food_title);
    $pars_food_weight = array_values($pars_food_weight);
    $pars_food_description = array_values($pars_food_description);
    $pars_food_price = array_values($pars_food_price);
    $selectors_del = array_values($selectors_del);
    $category_url = array_values($category_url);
    $category_id = array_values($category_id);
    $img_url = array_values($img_url);
    $safe_id = array_values($safe_id);
    //парсинг
for ($i=0; $i < count($category_url); $i++) {
    $html[$i] = Parsing::get_page_by_url($category_url[$i]);    $dom[$i]
phpQuery::newDocument($html[$i]);
    if (!empty($selector_delete[$i])) {
        for ($i=0; $i < count($selector_delete); $i++) {
            $selectors_del[$i] = explode(",", $selector_delete[$i]);
        }
    }
    /* - обязательные значения
foreach ($dom[$i]->find($pars_product[$i]) as $key[$i] => $value[$i]) {
    $pq[$i] = pq($value[$i]);
    //Изображения блюда
    if (!empty($pars_food_img[$i])) {
        $val[$i]['food_img'] = $pq[$i]->find($pars_food_img[$i])->attr('src');
    }
    else {
        $val[$i]['food_img'] = "../img/empty.png";
    }
    //Название блюда*
    $val[$i]['food_title'] = $pq[$i]->find($pars_food_title[$i])->text();
    //Вес/кол-ство
    if (!empty($pars_food_weight[$i])) {
        $val[$i]['weight'] = $pq[$i]->find($pars_food_weight[$i])->text();
    }
    else {
        $val[$i]['weight'] = " - ";
    }
}

```

```

    }
    //Состав блюда
    if (!empty($pars_food_description[$i])) {
        $val[$i]['food_description'] = $pq[$i]->find($pars_food_description[$i])->text();
    }
    else {
        $val[$i]['food_description'] = " - ";
    }
    //Стоимость блюда*
    $val[$i]['price'] = $pq[$i]->find($pars_food_price[$i])->text();
    //Url категории*
    $val[$i]['category_url'] = $category_url[$i];
    //ID категории*
    $val[$i]['category_id'] = $category_id[$i];
    //Url img
    if (!empty($pars_food_description[$i])) {
        $val[$i]['img_url'] = $img_url[$i];
    }
    else {
        $val[$i]['img_url'] = "";
    }
    //ID кафе*
    $val[$i]['cafe_id'] = $cafe_id[$i];
    array_push($vals, $val[$i]);
}

}

$columns = DB::getColomns('food');
$keys = array_values($columns);
array_shift($keys); //извлекает первое значение массива array и возвращает его, сокращая
размер array на один элемент
//использование этого метода для того чтобы не брать id
$table = "food";
for ($j=0; $j < count($vals); $j++) {
    DB::insert($keys, $vals[$j], $table);
}

?>

```