#### МУНИЦИПАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

#### «СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА №5»

#### 

Исследовательская работа на тему:

**«Язык Web-программирования - PHP»**

Выполнил: учащийся

9 «А» класса

Рябов Артем

Педагог: Абдразакова А.К.

Когалым

2011 г

СОДЕРЖАНИЕ

ВВЕДЕНИЕ………………………………………………………………………………….…...3

**Глава 1. Вступление в PHP и MySQL………………………..……………..…….………….5**

**1.1. Что такое PHP?.………………………………………………………………………5**

**1.2. Возможности PHP…………………………………………………………………….5**

**1.3. Краткая история PHP……….…...…………………….……………………...…......6**

**1.4. Почему нужно выбирать PHP………………………………………………………7**

**1.5. Что такое MySQL……………………………………………………………………..7**

**1.6. Возможности MySQL…………………………………………………………………8**

**Глава 2. Примеры использования PHP……………………………………………………...9**

**2.1. Работа с формами……………………………………………………………………..9**

**2.2. Работа с MySQL (сохранение данных в базе данных)...…………………………11**

**2.3. Работа с MySQL (получение данных из базы данных).…………………………13**

ЗАКЛЮЧЕНИЕ………………………………………………………………………..……….15

БИБЛИОГРАФИЧЕСКИЙ СПИСОК …………………………………………………...…16

ПРИЛОЖЕНИЯ………………………………………………………………………………...17

**Введение**

PHP - язык программирования, используемый на стороне WEB-сервера для динамической генерации HTML-страниц. Об этом говорит и расшифровка его названия: PHP - Personal HyperText Processor.

PHP - один из немногих языков программирования, созданных специально для разработки веб-приложений. Поэтому он включает в себя все функции, необходимые именно для работы на веб-сервере, и при этом лишен избыточности, свойственной многим его конкурентам.

Очень приятная особенность PHP - то, что его команды включаются в обычные HTML-страницы с помощью специальных тегов, которые и заставляют PHP-машину выполнять на сервере нужные действия. Программам на PHP не нужны специальные CGI-директории с особыми правами доступа. Более того, на одной страничке можно произвольно чередовать "простой" HTML и PHP-код.

PHP не зависит от платформы. PHP прекрасно интегрируется во все популярные веб-серверы: Apacce и IIS, Zens и Netscape Enterprise Server, работает под Windows и OS/2, MacOS и практически всеми UNIX-подобными системами. Как следствие - PHP работает практически у всех хостеров, разрешающих собственные выполняемые скрипты.

Замечательная особенность PHP - его интегрированность практически со всеми современными интернет-технологиями. PHP поддерживает большинство современных веб-протоколов: IMAP, FTP, POP, XML, SNMP и другие. PHP прекрасно работает с базами данных. Трудно найти СУБД, поддержка которой не была бы реализована в PHP. MySQL и MS SQL Server, PostgreSQL и Oracle, Sybase и Interbase... Один только перечень баз данных, поддерживаемых PHP, займет, наверное, целый экран.

PHP включает в себя огромное количество встроенных функций: обработки строк и массивов, работы с файловой системой и с HTTP, электронной почтой, датой и временем, кириллицей и другими национальными алфавитами... Когда я впервые начал программировать на PHP, то был просто поражен обилием встроенных функций! Благодаря им многие алгоритмы, требующие в большинстве языков написания программного кода размером в несколько экранов, реализуются на PHP одной командой (точнее, вызовом одной функции).

Современные тенденции развития языков программирования не обошли стороной и PHP. Средства объектно-ориентированного программирования появились еще в PHP3. А в объектной модели PHP4 в полном объеме реализованы классические понятия объектно-ориентированного программирования: наследование, инкапсуляция и полиморфизм.

Все вышеизложенное позволяет без всякой натяжки назвать PHP безусловным лидером среди языков веб-программирования.

**Цель** **исследования** – Изучить и посмотреть примеры выполнения скриптов PHP

**Объект** **исследования** – Язык PHP, Базы данных MySQL.

**Предмет исследования** – функциональное значение и актуальность языка.

**Гипотеза** **исследования** состоит в том, что данный язык очень простой, легко интегрируется в HTML, в связке PHP+MySQL+HTML намного превосходит простой HTML.

Исходя из гипотезы, сформированы следующие **задачи**:

**Задачи:**

* изучить особенности и возможности языка PHP;
* сравнить функционал PHP и HTML;
* познакомиться с базами данных MySQL;
* обработать полученные результаты, сделать выводы.

Теоретической основой представленного материала являются исторические данные, представленные в справочниках, материал из интернета.

**Методы исследования:** в процессе проведения исследований применялись описание, анализ литературы по теме, изучение некоторых переменных.

**Практическая значимость исследования**: данный материал можно использовать учителям и учащимся для последующего, углубленного изучения данного языка.

**Структура и содержание работы включают:**

Вводную часть, в которой обоснован выбор и актуальность темы.

Глава 1. «Вступление в PHP и MySQL» содержит 6 разделов. В первом разделе идет объяснение, что такое PHP. Во втором описаны его возможности. В третьем разделе главы описывается история языка PHP. В четвертом - сказано почему стоит выбирать именно этот язык программирования. Пятый и шестой разделы посвящены информации о базах данных MySQL.

**Глава 2. «Примеры использования PHP»** состоит из 3 разделов. В первом разделе показано как в PHP легко обрабатывать данные с HTML – форм. Во втором мы будем заносить информацию в базу данных. В третьем разделе будем получать информацию из баз данных.

**Заключение** содержит основные выводы по теме, подтверждающие выдвинутую гипотезу исследования.

**Глава 1.**  **Вступление в PHP и MySQL**

* 1. **Что такое PHP?**

PHP – это скрипт-язык (scripting language), встраиваемый в HTML, который интерпретируется и выполняется на сервере. Проще всего это показать на примере:

|  |
| --- |
| <html>  <head>  <title>Пример</title>  </head>  <body>  <?php echo "Hi, I'm a PHP script!"; ?>  </body>  </html> |

После выполнения этого скрипта мы получим страничку, в которой будет написано:

**Hi, I'm a PHP script!** (см. приложение 1)

Весьма просто и бесполезно.

Основное отличие от CGI-скриптов, написанных на других языках, типа Perl или C – это то, что в CGI-программах вы сами пишете выводимый HTML-код, а, используя PHP – вы встраиваете свою программу в готовую HTML-страницу, используя открывающий и закрывающий теги (в примере <?php и ?>).

Отличие PHP от JavaScript, состоит в том, что PHP-скрипт выполняется на сервере, а клиенту передается результат работы, тогда как в JavaScript-код полностью передается на клиентскую машину и только там выполняется.

Любители Internet Information Server найдут, что PHP очень похож на Active Server Pages (ASP), а энтузиасты Java скажут, что PHP похож на Java Server Pages (JSP). Все три языка позволяют размещать код, выполняемый на Web-сервере, внутри HTML страниц.

**1.2. Возможности PHP**

В нескольких словах – на PHP можно сделать все, что можно сделать с помощью CGI–программ. Например: обрабатывать данные из форм, генерировать динамические страницы, получать и посылать куки (cookies).

Кроме этого в PHP включена поддержка многих баз данных (databases), что делает написание Web-приложений с использованием БД до невозможности простым.

Вот неполный перечень поддерживаемых БД:

|  |  |  |
| --- | --- | --- |
| Adabas D | InterBase | Solid |
| dBase | mSQL | Sybase |
| Empress | MySQL | Velocis |
| FilePro | Oracle | Unix dbm |
| Informix | PostgreSQL |  |

Вдобавок ко всему PHP понимает протоколы IMAP, SNMP, NNTP, POP3 и даже HTTP, а также имеет возможность работать с сокетами (sockets) и общаться по другим протоколам.

**1.3. Краткая история PHP**

Началом PHP можно считать осень 1994 года, когда Расмус Лердорф (Rasmus Lerdorf) решил расширить возможности своей Home-page (Домашней страницы) и написать небольшой движок для выполнения простейших задач. Такой движок был готов к началу 1995 года и назывался Personal Home Page Tools. Умел он не очень много – понимал простейший язык и всего несколько макросов.

К середине 1995 года появилась вторая версия, которая называлась PHP/FI Version 2. Приставка FI – присоединилась из другого пакета Расмуса, который умел обрабатывать формы (Form Interpritator). PHP/FI компилировался внутрь Apache и использовал стандартный API Apache. PHP скрипты оказались быстрее аналогичных CGI – скриптов, так как серверу не было необходимости порождать новый процесс. Язык PHP по возможностям приблизился к Perl, самому популярному языку для написания CGI-программ. Была добавлена поддержка множества известных баз данных (например, MySQL и Oracle). Интерфейс к GD – библиотеке, позволял генерировать картинки на лету. С этого момента началось широкое распространение PHP/FI.

В конце 1997 Зээв Сураски (Zeev Suraski) и Энди Гутманс (Andi Gutmans) решили переписать внутренний движок, с целью исправить ошибки интерпретатора и повысить скорость выполнения скриптов. Через полгода, 6 июня 1998 года вышла новая версия, которая была названа PHP 3. К лету 1999 года PHP 3 был включен в несколько коммерческих продуктов. По данным NetCraft на ноябрь 1999 PHP использовался в более чем 1 млн. доменах.

**1.4. Почему нужно выбирать PHP**

Разработчикам Web-приложений нет необходимости говорить, что web-страницы - это не только текст и картинки. Достойный внимания сайт должен поддерживать некоторый уровень интерактивности с пользователем : поиск информации, продажа продуктов, конференции и т.п. Традиционно все это реализовалось CGI-скриптами, написанными на Perl. Но CGI- скрипты очень плохо масштабируемы. Каждый новый вызов CGI, требует от ядра порождения нового процесса, а это занимает процессорное время и тратит оперативную память. PHP предлагает другой вариант – он работает как часть Web-сервера, и этим самым похож на ASP от Microsoft.

Синтаксис PHP очень похож на синтаксис C или Perl. Люди, знакомые с программированием, очень быстро смогут начать писать программы на PHP. В этом языке нет строгой типизации данных и нет необходимости в действиях по выделению/освобождению памяти.

Программы, написанные на PHP, достаточно легкочитаемы. Написанный PHP – код легко зрительно прочитать и понять, в отличие от Perl-программ.

Недостатки PHP

* PHP является интерпретируемым языком, и, вследствие этого, не может сравниться по скорости с компилируемым С. Однако при написании небольших программ, что, в общем-то, присуще проектам на PHP, когда весь проект состоит из многих небольших страниц с кодом, вступают в силу накладные расходы на загрузку в память и вызов CGI-программы, написанной на С.
* Не такая большая база готовых модулей, как, например, СPAN у Perl. С этим ничего нельзя поделать – это дело времени. В PHP 4 разработчики предусмотрели специальный репозиторий PEAR, аналогичный CPAN, и я думаю, очень скоро будет написано достаточное количество модулей для его наполнения.

**1.5. Что такое MySQL**

MySQL – компактный многопоточный сервер баз данных. MySQL характеризуется большой скоростью, устойчивостью и легкостью в использовании.

MySQL был разработан компанией TcX для внутренних нужд, которые заключались в быстрой обработке очень больших баз данных. Компания утверждает, что использует MySQL с 1996 года на сервере с более чем 40 БД, которые содержат 10,000 таблиц, из которых более чем 500 имеют более 7 миллионов строк.

MySQL является идеальным решением для малых и средних приложений. Исходники сервера компилируются на множестве платформ. Наиболее полно возможности сервера проявляются на Unix-серверах, где есть поддержка многопоточности, что дает значительный прирост производительности.

На текущий момент MySQL все еще в стадии разработки, хотя версии 3.22 полностью работоспособны.

MySQL-сервер является бесплатным для некоммерческого использования. Иначе необходимо приобретение лицензии, стоимость которой составляет 190 EUR.

**1.6. Возможности MySQL**

MySQL поддерживает язык запросов SQL в стандарте ANSI 92, и кроме этого имеет множество расширений к этому стандарту, которых нет ни в одной другой СУБД.

Краткий перечень возможностей MySQL.

* Поддерживается неограниченное количество пользователей, одновременно работающих с базой данных.
* Количество строк в таблицах может достигать 50 млн.
* Быстрое выполнение команд. Возможно MySQL самый быстрый сервер из существующих.
* Простая и эффективная система безопасности.

MySQL действительно очень быстрый сервер, но для достижения этого разработчикам пришлось пожертвовать некоторыми требованиями к реляционным СУБД.

В MySQL отсутствуют:

* Поддержка вложенных запросов, типа SELECT \* FROM table1 WHERE id IN (SELECT id FROM table2).
* Не реализована поддержка транзакций. Взамен предлагается использовать LOCK/UNLOCK TABLE.
* Нет поддержки триггеров и хранимых процедур.

По словам создателей именно эти пункты дали возможность достичь высокого быстродействия. Их реализация существенно снижает скорость сервера. Эти возможности не являются критичными при создании Web-приложений, что в сочетании с высоким быстродействием и малой ценой позволило серверу приобрести большую популярность

**Глава 2. Примеры использования PHP**

**2.1. Работа с формами**

В этом примере показано как в PHP легко обрабатывать данные с HTML – форм.

Создадим простой HTML файл.

|  |
| --- |
| <HTML>  <HEAD>  <TITLE>Запрос информации</TITLE>  <BODY>  <CENTER>  Хотите больше знать о наших товарах?  <P>  <TABLE WIDTH = 400><TR><TD align = right>  <FORM ACTION="email.php" METHOD="POST">  Ваше имя:<BR> <INPUT TYPE="text" NAME="name" SIZE="20" MAXLENGTH="30">  <P>  Ваш email:<BR> <INPUT TYPE="text" NAME="email" SIZE="20" MAXLENGTH="30">  <P>  Меня интересуют:  <SELECT NAME="preference">  <OPTION value = "Яблоки">Яблоки  <OPTION value = "Апельсины">Апельсины  </SELECT>  <P>  <INPUT TYPE="submit" VALUE="Отправить запрос!">  </FORM>  </TD></TR></TABLE></CENTER>  </BODY>  </HTML> |

Назовем этот файл request.html. В нем мы указали, что данные формы будут обрабатываться файлом email.php. Приведем его содержание:

|  |
| --- |
| <?  /\* Этот скрипт получает переменные из request.html \*/  PRINT "<CENTER>";  PRINT "Привет, ".$\_POST['name'];  PRINT "<BR><BR>";  PRINT "Спасибо за ваш интерес.<BR><BR>";  PRINT "Вас интересуют ".$\_POST['preference'].",  информацию о них мы пошлем вам на email: ".$\_POST['email']; PRINT "</CENTER>";  ?> |

Теперь, если пользователь вызовет request.html и наберет в форме имя “Вася”, email: vasya@pupkin.com и скажет, что его интересуют “Яблоки”, а после этого нажмет "Отправить запрос!", то в ответ вызовется email.php, который выведет на экран примерно следующее:

|  |
| --- |
| **Привет, Вася**  **Спасибо за ваш интерес.**  **Вас интересуют Яблоки. Информацию о них мы пошлем вам на email: vasya@pupkin.com** |

(Приложения 2, 3)

Теперь мы должны сдержать обещание и выслать email.

Для этого в PHP есть функция MAIL.

**Синтаксис: void mail(string to, string subject, string message, string add\_headers);**

to – email адрес получателя. subject – тема письма. message – собственно текст сообщения.

add\_headers – другие параметры заголовка письма (необязательный параметр).

Допишем в конец файла email.php следующий код:

|  |
| --- |
| <?php  $subj = "Запрос на информацию";  $text = "Уважаемый ".$\_POST['name']."!  Спасибо за ваш интерес!  Вас интересуют ".$\_POST['preference']."  Мы их распространяем бесплатно.  Обратитесь в ближайший филиал нашей компании и получите ящик этого продукта.";  mail($\_POST['email'], $subj, $text);  $subj = "Поступил запрос на информацию";  $text = $\_POST['name']." интересовали ".$\_POST['preference']." email-адрес: ".$\_POST['email'];  mail($adminaddress, $subj, $text);  ?> |

Вот теперь пользователь будет получать письмо с более подробной информацией о наших товарах. Также письмо получит и администратор сайта. (Приложение 4)

Когда интересующихся нашими товарами станет очень много, мы захотим их как-то упорядочить и хранить информацию о них в базе данных. Как раз это мы будем разбирать в следующем разделе.

**2.2. Работа с MySQL (сохранение данных в базе данных)**

Для начала создаем базу данных и таблицу. Входим в phpMyAdmin (phpMyAdmin — веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL) (см. приложение 5), и выполняем команды:

>CREATE DATABASE products;

>CREATE TABLE clients (name VARCHAR(25), email VARCHAR(25), choise VARCHAR(8));

Для общения с MySQL из PHP понадобятся следующие функции.

**int mysql\_connect(string hostname, string username, string password);**

Создать соединение с MySQL. Параметры: Hostname – имя хоста, на котором находится база данных.Username – имя пользователя.Password – пароль пользователя.

Функция возвращает параметр типа int, который больше 0, если соединение прошло успешно, и равен 0 в противном случае.

**int mysql\_select\_db(string database\_name, int link\_identifier);**

Выбрать базу данных для работы. Параметры:Database\_name – имя базы данных.link\_identifier – ID соединения, которое получено в функции mysql\_connect. (параметр необязательный, если он не указывается, то используется ID от последнего вызова mysql\_connect)

Функция возвращает значение true или false

**int mysql\_query(string query, int link\_identifier);**

Функция выполняет запрос к базе данных.Параметры:Query – строка, содержащая запросlink\_identifier – см. предыдущую функцию.

Функция возвращает ID результата или 0, если произошла ошибка.

**int mysql\_close(int link\_identifier);**

Функция закрывает соединение с MySQL.

Параметры:

link\_identifier – см. выше.

Функция возвращает значение true или false

Теперь наш файл email.php будет иметь след. вид:

|  |
| --- |
| <?  /\* Этот скрипт получает переменные из request.html \*/  /\* Переменные для соединения с базой данных \*/  $hostname = "localhost";  $username = "myusername";  $password = "mypassword";  $dbName = "products";  /\* Таблица MySQL, в которой хранятся данные \*/  $userstable = "clients";  /\* email администратора \*/  $adminaddress = "administration@me.com";  /\* создать соединение \*/  mysql\_connect($hostname,$username,$password) OR DIE("Не могу создать соединение ");  /\* выбрать базу данных. Если произойдет ошибка - вывести ее \*/  mysql\_select\_db($dbName) or die(mysql\_error());  echo "<CENTER>";  echo "Привет, ".$\_POST['name'];  echo "<BR><BR>";  echo "Спасибо за ваш интерес.<BR><BR>";  echo "Вас интересуют ".$\_POST['preference'].". Информацию о них мы пошлем вам на email: ".$\_POST['email'];  echo "</CENTER>";  /\* Отправляем email-ы \*/  $subj = "Запрос на информацию";  $text = "Уважаемый ".$\_POST['name']."!  Спасибо за ваш интерес!  Вас интересуют ".$\_POST['preference']."  Мы их распространяем бесплатно.  Обратитесь в ближайший филиал нашей компании и получите ящик этого продукта.";  mail($\_POST['email'], $subj, $text);  $subj="Поступил запрос на информацию";  $text = $\_POST['name']." интересовали ".$\_POST['preference']." email-адрес: ".$\_POST['email'];  mail($adminaddress, $subj, $text);  /\* составить запрос для вставки информации о клиенте в таблицу \*/  $query = "INSERT INTO $userstable VALUES('$name','$email', '$preference')";  /\* Выполнить запрос. Если произойдет ошибка - вывести ее. \*/  mysql\_query($query) or die(mysql\_error());  echo "Информация о вас занесена в базу данных.";  /\* Закрыть соединение \*/  mysql\_close();  ?> |

(Приложение 6)

Вот так легко можно работать с базой данных в PHP. Теперь кроме письменных уведомлений, информация о клиенте и его интересах будет заносится в таблицу MySQL.

**2.3. Работа с MySQL (получение данных из базы данных)**

После занесения данных, нас иногда будет интересовать вопрос так кого же из наших клиентов интересует товар “Яблоки” (не путать с Apple Macintosh, по поводу Apple Macintosh см. www.stealthcomp.com).

Напишем скрипт apple.php

|  |
| --- |
| <?  /\* Скрипт показывает клиентов, которые яблоки любят больше чем апельсины \*/  /\* Переменные для соединения с базой данных \*/  $hostname = "localhost";  $username = "myusername";  $password = "mypassword";  $dbName = "products";  /\* Таблица MySQL, в которой хранятся данные \*/  $userstable = "clients";  /\* создать соединение \*/  mysql\_connect($hostname,$username,$password) OR DIE("Не могу создать соединение ");  /\* выбрать базу данных. Если произойдет ошибка - вывести ее \*/  mysql\_select\_db($dbName) or die(mysql\_error());  /\* составить запрос, который выберет всех клиентов - яблочников \*/  $query = "SELECT \* FROM $userstable WHERE choise = 'Яблоки'";  /\* Выполнить запрос. Если произойдет ошибка - вывести ее. \*/  $res = mysql\_query($query) or die(mysql\_error());  /\* Как много нашлось таких \*/  $number = mysql\_num\_rows($res);  /\* Напечатать всех в красивом виде\*/  if ($number == 0) {  echo "<CENTER><P>Любителей яблок нет</CENTER>";  } else {  echo "<CENTER><P>Количество любителей яблок: $number<BR><BR>";  /\* Получать по одной строке из таблицы в массив $row, пока строки не кончатся \*/  while ($row=mysql\_fetch\_array($res)) {  echo "Клиент ".$row['name']." любит Яблоки.<BR>";  echo "Его Email: ".$row['email'];  echo "<BR><BR>";  }  echo "</CENTER>";  }  ?> |

Вот и все, коммерческий продукт практически готов.

**Заключение**

В результате изучении темы и проведенных сравнений, выдвинутая мной гипотеза была подтверждена.

Можно сделать вывод о том, что сайты с использованием связки HTML+PHP+MySQL в несколько раз превосходят обычные HTML сайты по функциональности, удобству и т.д.

Неудивительно, что эта связка PHP+MySQL развивается очень быстро. Постоянно появляются новые возможности, разные полезные скрипты, которые позволяют усовершенствовать свой сайт.

Различные технологи вроде Flash, Ajax, JavaScript хорошо взаимодействуют с PHP, что позволяет создавать просто сумасшедшие по функциональности и красоте сайты.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

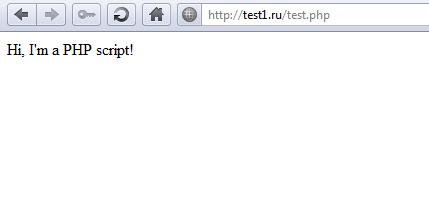
1. Джордж Шлосснейгл. Профессиональное программирование на PHP, 2006
2. Уильям Стейнмец, Брайан Вард. 75 готовых решений для вашего web-сайта на РНР, 2009
3. Мишель Е. Дэвис и Джон А. Филипс. Изучаем PHP и MySQL, 2008
4. Николай Прохоренок. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера, 2010
5. Тим Конверс, Джойс Парк и Кларк Морган. PHP 5 и MySQL. Библия пользователя, 2006

Электронные адреса:

1. <http://www.php.su/>
2. <http://www.mirsite.ru/>
3. <http://www.master-live.ru/php_klass.php>
4. <http://www.sitescript.ru/>
5. <http://ruseller.com/>

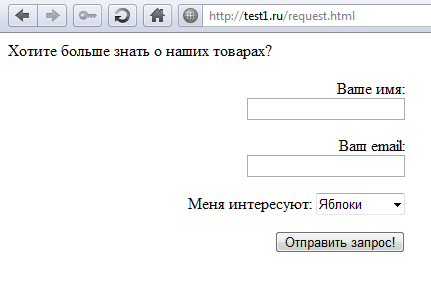
**ПРИЛОЖЕНИЯ**

**Приложение 1**



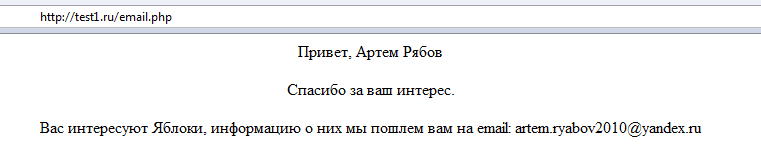
Пример вывода текста скрипта в браузер

**Приложение 2**



Форма заявки

**Приложение 3**



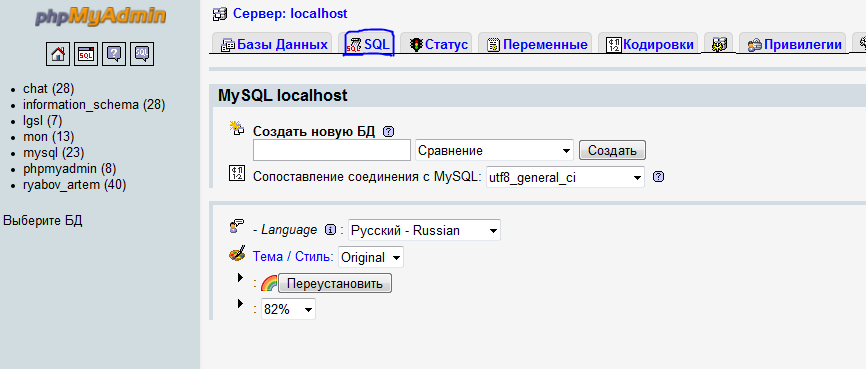
Вывод сообщения пользователю, интересующемуся товаром «Яблоки»

**Приложение 4**



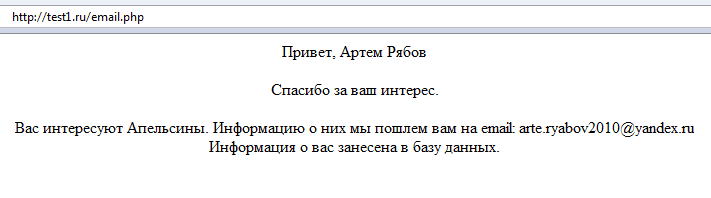
Электронное письмо, отправившееся пользователю

**Приложение 5**



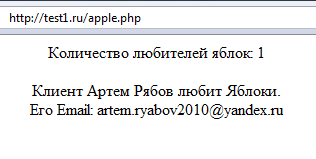
phpMyAdmin — веб-приложение с открытым кодом, написанное на языке PHP и представляющее собой веб-интерфейс для администрирования СУБД MySQL

**Приложение 6**



Измененный вывод сообщения пользователю, интересующимуся товаром «Яблоки»

**Приложение 7**



Вывод информации о количестве пользователей, которым нравится тот или иной товар