



Панферова Валерия Сергеевна, Мурашев Игорь Дмитриевич,

Российский биотехнологический университет (РОСБИОТЕХ)

Panferova Valeria Sergeevna, Murashev Igor Dmitrievich,

Russian Biotechnological University (ROSBIOTECH)

**РАЗРАБОТКА СИСТЕМЫ КОНТРОЛЯ ИНГРЕДИЕНТОВ ВЕБ-САЙТА
С РЕЦЕПТАМИ НА PHP С БАЗОЙ ДАННЫХ MYSQL
С ИСПОЛЬЗОВАНИЕМ HTML И CSS
DEVELOPMENT OF AN INGREDIENT CONTROL SYSTEM FOR
A WEBSITE WITH RECIPES IN PHP WITH A MYSQL DATABASE
USING HTML AND CSS**

Аннотация: В работе рассмотрен процесс создания системы для обновления информации в базе данных MySQL ингредиентов к веб - сайту с рецептами. Было созданы формы для добавления, удаления и изменения ингредиентов, а также реализован поиск в базе данных с помощью языка программирования PHP.

Abstract: The paper considers the process of creating a system for updating information in the MySQL database of ingredients to a website with recipes. Forms were created for adding, removing and changing ingredients, and a database search was implemented using the PHP programming language.

Ключевые слова: PHP, HTML, CSS, MySQL, локальный сервер, ингредиенты.

Keywords: PHP, HTML, CSS, MySQL, local server, ingredients.

Пищевая тематика в любой период времени будет популярна, так как пища - это одна из приоритетных потребностей каждой живой сущности [1]. Исходя из этого были придуманы рецепты различных видов для более полезного, а также вкусного потребления пищи. Каждый рецепт состоит из



каких-либо ингредиентов. В нашей работе было рассмотрено создание фрагмента базы данных для веб-сайта с рецептами с использованием базы данных MySQL, написание некоторой системы для взаимодействия с данной базой данных, а именно добавление ингредиентов в базу, поиск и вывод ингредиентов из базы, удаление или изменение ингредиентов в базе [2]. Также был реализован дизайн веб-сайта с использованием языка разметки HTML и языка стилей CSS [3] [4].

Для реализации данного проекта был использован локальный сервер OpenServer [5]. Данное программное обеспечение является бесплатным и лежит в открытом доступе в интернете. На локальном сервере была создана база данных MySQL для реализации веб-проекта с рецептами. Но в данной работе был рассмотрен лишь фрагмент от всей базы данных, который представлен на рисунке 1 [6].

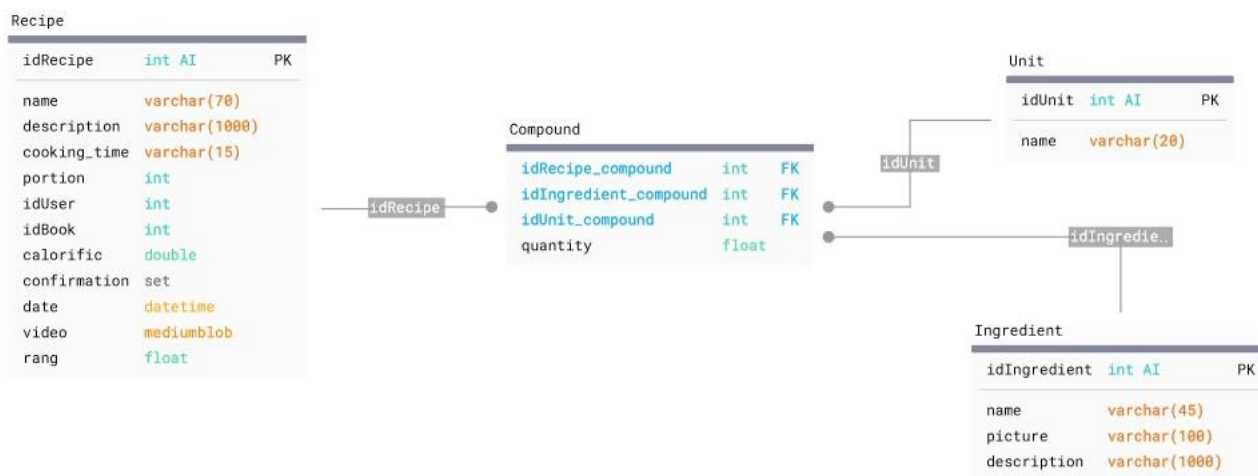


Рис. 1 - Фрагмент базы данных системы веб-сайта рецептов

На рисунке 1 изображены таблицы:

- Recipe. Содержит в себе информацию о рецепте.
- Compound. Является таблицей, где содержится состав рецепта, то есть ингредиенты, содержащиеся в рецепте.
- Unit. В таблице Unit содержится информация о единицах измерения ингредиента при указании ингредиента в рецепте.



- Ingredient. Содержит информацию об ингредиентах в базе данных.

После первичной установки и настройки OpenServer нам необходимо реализовать код для работы нашей системы. В качестве языка программирования был выбран язык PHP. С помощью него были реализованы запросы к созданной базе данных и вся системная часть веб-сайта. В качестве реализации дизайна веб-сайта был выбран язык разметки HTML и язык стилей CSS.

Реализация взаимодействия с базой данных MySQL будет происходить на 1 странице. В первую очередь нам необходимо создать форму для добавления ингредиента в базу данных. Для этого на главной странице update.php мы реализуем код, который отображён на листинге 1.

Листинг 1

```
update.php x
162
163 <!-- Добавить ингредиент -->
164
165 <form class="add-ingr-form" action="ingradd.php" method="post" enctype="multipart/form-data">
166 <h3>Добавление ингредиента</h3>
167 <?php if (isset($_SESSION['ingraddphoto'])) { ?>
168 <div class="message-block"> <?php echo $_SESSION['ingraddphoto']; unset($_SESSION['ingraddphoto']); ?></div>
169 <?php }; ?>
170
171 <div class="img">
172 
173 </div>
174 <label for="imgInp" class="input-file">
175 <input accept="image/*" type="file" name="file" id="imgInp" >
176 </label>
177 <div class="subtitle-img">
178 Внимание! Добавляйте изображение с соответствующим названием ингредиента. Например: Банан - banan.
179 </div>
180 <label for="title">Название ингредиента (не более 45 знаков). Это поле не может быть пустым.</label>
181 <?php if (isset($_SESSION['ingraddtitle'])) { ?>
182 <div class="message-block"> <?php echo $_SESSION['ingraddtitle']; unset($_SESSION['ingraddtitle']); ?></div>
183 <?php }; ?>
184 <input type="text" name="title" value="">
185 <label for="description">Описание ингредиента (не более 1000 знаков). Это поле не может быть пустым.</label>
186 <?php if (isset($_SESSION['ingradddescription'])) { ?>
187 <div class="message-block"> <?php echo $_SESSION['ingradddescription']; unset($_SESSION['ingradddescription']); ?></div>
188 <?php }; ?>
189 <textarea class="description-ingr" type="text" name="description" value=""></textarea>
190 <input type="submit" name="submit-ingradd" value="Добавить">
191 </form>
```

На листинге 1 форма для добавления ингредиента в базу данных имеет такие поля, как:

- Изображение ингредиента;



- Название ингредиента;
- Описание ингредиента.

Все поля в данной форме являются обязательными. В случае ошибки в том или ином поле будет выводиться определённая ячейка из глобального массива \$_SESSION с указанием на совершённую ошибку пользователем [7].

Контрольный пример формы для добавления ингредиента в базу данных отображён на рисунке 2.

Добавление ингредиента

Выберите файл | Файл не выбран

Внимание! Добавляйте изображение с соответствующим названием ингредиента. Например: Банан - banana.

Название ингредиента (не более 45 знаков). Это поле не может быть пустым.

Описание ингредиента (не более 1000 знаков). Это поле не может быть пустым.

Добавить

Рис. 2 - Контрольный пример формы добавления ингредиента в базу данных

Были добавлены указания при заполнении данных полей ввода, но даже при указании требований необходимо провести проверку на корректность введённых данных. Для проверки полей был создан новый файл `ingradd.php`. Код файла `ingradd.php` представлен на листинге 2 и листинге 3.



```
ingradd.php      x      update.php      x
1  <?php
2  //Добавление ингредиента в режиме модератора
3  session_start();
4  include ("config/connect.php");
5  function validate($data) {
6      $data=trim($data);
7      $data=stripslashes($data);
8      return $data;
9  }
10 //Обработка
11 $title=validate($_POST['title']);
12 $description=validate($_POST['description']);
13
14 //Загрузка фотографии ингредиента
15 if(!empty($_FILES['file'])) {
16     $file=$_FILES['file'];
17     $filename=$file['name'];
18     $pathFile=__DIR__.'./ingredients/'.$filename;
19
20     if(!move_uploaded_file($file['tmp_name'], $pathFile)) {
21         echo'Файл не смог загрузиться';
22     }
23 }
24 if($filename==NULL) {
25     $_SESSION['ingradd1']=false;
26     $_SESSION['ingraddphoto']='Обязательно надо добавить фотографию';
27     header('Location: update.php');
28 }else {
29     $_SESSION['ingradd1']=true;
30     unset($_SESSION['ingraddphoto']);
31 };
```

На листинге 2 отображён фрагмент кода для проверки введённых данных пользователем. Для корректировки текстовых данных была создана такая функция, как `validate()`. Перед записью в переменные, текстовые данные проходят обработку в данной функции. Она убирает лишние пробелы спереди и сзади строки (любое количество), а также исключает экранирование символов. Также осуществляется загрузка фотографии (изображения) ингредиента. В



случае корректного загружаемого файла, файл с изображением записывается в определённую директорию на сервере (директория указывается в переменной `$pathFile`). В случае ошибки загрузки файла будет выводиться соответствующее сообщение.

Листинг 3

```
ingradd.php x update.php x
35 //Проверка названия ингредиента
36 if(strlen($title)>45 || empty($title)) {
37     $_SESSION['ingraddtitle']='Вы превысили допустимый предел по количеству символов. У вас ' . strlen($title) . ' символов.';
38     $_SESSION['ingradd2']=false;
39     header('Location: update.php');
40 } else {
41     $_SESSION['ingradd2']=true;
42     unset($_SESSION['ingraddtitle']);
43 }
44 //Проверка описания ингредиента
45 if(strlen($description)>1000||empty($description)) {
46     $_SESSION['ingradddescription']='Вы превысили допустимый предел по количеству символов. У вас ' . strlen($description) . ' символов.';
47     $_SESSION['ingradd3']=false;
48     header('Location: update.php');
49 } else {
50     $_SESSION['ingradd3']=true;
51     unset($_SESSION['ingradddescription']);
52 }
53 //Загрузка на сервер
54 if($_SESSION['ingradd1']==true && $_SESSION['ingradd2']==true && $_SESSION['ingradd3']==true) {
55     $sql="INSERT INTO `Ingredient` (`name`,`picture`,`description`) VALUES ('$title','$filename','$description')";
56     mysqli_query($connect,$sql);
57     $_SESSION['ok']='Ингредиент успешно добавлен';
58     unset($_SESSION['notok']);
59     header('Location: update.php');
60 } else {
61     $_SESSION['notok']='Что-то пошло не так';
62     unset($_SESSION['ok']);
63     header('Location: update.php');
64 }
65 ?>
```

На листинге 3 отображена проверка названия ингредиента, что оно существует и не превышает 45 символов. Также показана проверка описания ингредиента, что оно существует и не превышает 1000 символов. В случае ошибок, сообщение об ошибке записывается в определённый элемент массива `$_SESSION` и выводится на экране с формой на заполнение. Далее производится загрузка на сервер. В случае успешном прохождении проверок, информация об ингредиенте добавляется в таблицу. В качестве картинки в базу данных записывается только название изображения [8]. Если проверка не пройдена, то выводится сообщение и данные не записываются в базу данных.

Затем, нам необходимо сделать изменение ингредиента. Для изменения определённого ингредиента был реализован поиск в таблице. Реализация формы ввода в поиске отображена в листинге 4.



```
update.php x
198 <!--Изменить ингредиент-->
199
200 <div class="ingr-up-search-block">
201 <h3 class="title">Изменить ингредиент</h3>
202 <form id="srresup" class="" action="<?=$_SERVER['SCRIPT_NAME'] ?>#srresup">
203   <input type="text" name="search-ingr-up" value="" placeholder="Введите название ингредиента">
204   <input type="submit" name="submit-search-ingr-up" value="Найти">
205 </form>
206 <div class="ingr-list">
207 <?php
208 if(isset($inputSearch)) {
209   countPeople($result); //Функция вывода ингредиентов
210 } else {
211   echo "<div class='search-text'>Здесь отображается список ингредиентов.</div>";
212 }
213 ?>
214 </div>
215 </div>
```

На листинге 4 показана реализация поиска для изменения ингредиента. Пользователь вводит название ингредиента в поле ввода. По нажатию пользователем кнопки «Найти» страница выводит список ингредиентов, если таковые есть, т. е. вызывается функция `someIngredients()`. Код поиска представлен на листинге 5.

```
update.php x
8 if(isset($_REQUEST['search-ingr-up'])) {
9   $inputSearch=$_REQUEST['search-ingr-up'];
10  if($inputSearch!=NULL) {
11    if((strlen($inputSearch)/2)==1) {
12      $symbol1=mb_substr($inputSearch,0);
13      $sql="SELECT * FROM `Ingredient` WHERE `name`='$inputSearch' OR (`name` LIKE '{$symbol1}%') ORDER BY `name` ASC";
14      $result=mysqli_query($connect,$sql);
15    } else {
16      $symbol3=mb_substr($inputSearch,0,3);
17      $sql="SELECT * FROM `Ingredient` WHERE `name`='$inputSearch' OR (`name` LIKE '%{$symbol3}%')";
18      $result=mysqli_query($connect,$sql);
19    }
20  }
21 } else {
22   $sql="SELECT * FROM `Ingredient` WHERE `name`='$inputSearch'";
23   $result=mysqli_query($connect,$sql);
24 }
25 }
```

На листинге 5 показана реализация поиска по базе данных [9]. Поиск был реализован с помощью запросов в MySQL. В данном случае выводится список ингредиентов по названию, где существуют совпадения с введённой строкой пользователя по первому символу или по трём. Список ингредиентов отсортирован по возрастанию. Контрольный пример показан на рисунке 3.

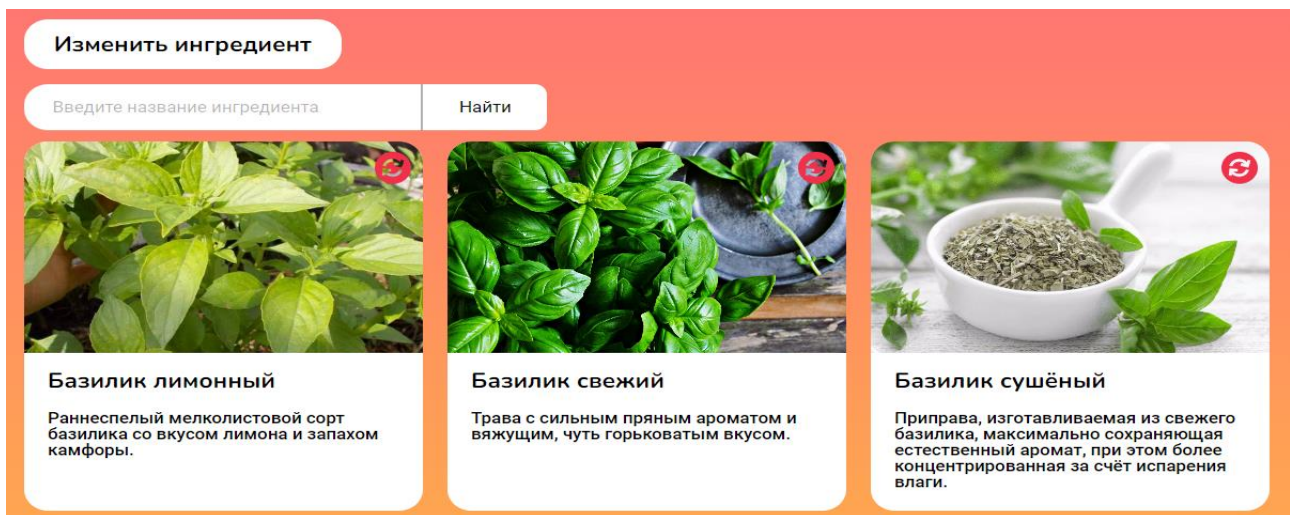


Рис. 3 - Контрольный пример вывода ингредиентов при поиске

На контрольном примере выводятся карточки ингредиентов по 3 в ряд. Отображены изображения ингредиентов, их название и краткое описание к ним.

Далее при нажатии кнопки на карточке ингредиента в правом верхнем углу открывается форма для изменения данных ингредиента. Код формы представлен на листинге 6.

Листинг 6

```
update.php
219 <form id="srup" class="up-ingr-form" <?php if(isset($_GET['id'])) {echo "show";} else {echo "hide";} ?>" action="ingrup.php" method="post">
220 <?php
221 if(isset($_GET['id'])||isset($_SESSION['idIngredient'])) {
222     if(isset($_GET['id']) && $_GET['id']!=NULL) {
223         $_SESSION['idIngredient']=$_GET['id'];
224         $updateIngredient=$_SESSION['idIngredient'];
225         $sql="SELECT * FROM `Ingredient` WHERE `idIngredient`='{$updateIngredient}'";
226         $result=mysqli_query($connect,$sql);
227         $Ingredient=mysqli_fetch_assoc($result);
228     }?>
229 <h3>Изменение ингредиента</h3>
230 <div class="img">
231     
232 </div>
233 <label for="imgInp" class="input-file">
234 <input accept="image/*" type="file" name="file" id="imgInp1" >
235 </label>
236 <div class="subtitle-img">Внимание! Добавляйте изображение с соответствующим названием ингредиента. Например: Банан - banan.</div>
237 <?php if (isset($_SESSION['upingname'])) { ?>
238 <div class="message-block"> <?php echo $_SESSION['upingname']; unset($_SESSION['upingname']); ?></div>
239 <?php ?>
240 <label for="name">Название ингредиента (не более 45 знаков). Это поле не может быть пустым.</label>
241 <input type="text" name="name" value="<?php if(isset($_GET['id'])) { echo $Ingredient['name'];} ?>">
242 <?php if (isset($_SESSION['upingdescription'])) { ?>
243 <div class="message-block"> <?php echo $_SESSION['upingdescription']; unset($_SESSION['upingdescription']); ?></div>
244 <?php ?>
245 <label for="description">Описание ингредиента (не более 1000 знаков). Это поле не может быть пустым.</label>
246 <textarea class="description-ingr" type="text" name="description" value=""><?php if(isset($_GET['id'])) {echo $Ingredient['description'];} ?></textarea>
247 <input type="submit" name="submit-ingrup" value="Обновить">
248 </form>
```

На листинге 6 показан код формы для обновления информации об выбранном ингредиенте. С помощью глобального массива `$_GET` мы берём информацию об ID выбранного ингредиента и выполняем запрос на вывод остальной информации об ингредиенте [10]. Полученные данные отображаем в



полях ввода информации в форме. Проверка полей происходит аналогично листингам 2 и 3. На рисунке 4 отображён контрольный пример формы для обновления информации об выбранном ингредиенте.

Изменение ингредиента



Внимание! Добавляйте изображение с соответствующим названием ингредиента. Например: Банан - banan.

Название ингредиента (не более 45 знаков). Это поле не может быть пустым.

Описание ингредиента (не более 1000 знаков). Это поле не может быть пустым.

Трава с сильным пряным ароматом и вяжущим, чуть горьковатым вкусом.

Рис. 4 - Контрольный пример формы для изменения ингредиента

После изменения информации в полях ввода, нам необходимо отправить запрос на сервер для изменения информации в базе данных об выбранном ингредиенте. Для реализации проверки данных, а также отправки запроса на сервер был создан файл `ingrup.php`. Отправка запроса в базу данных реализована в листинге 7.

Листинг 7

```
ingrup.php x update.php x
49 if($_SESSION['ingrup1']==true && $_SESSION['ingrup2']==true) {
50     $sql="UPDATE `Ingredient` SET `name`='{${name}',$ `picture`='{${filename}',$ `description`='{${description}'
51     WHERE `idIngredient`='{${Ingredient['idIngredient']}'";
52     mysqli_query($connect,$sql);
53     $_SESSION['ok']='Ингредиент успешно изменён';
54     header('Location: update.php');
55 } else {
56     unset($_SESSION['ok']);
57 }
58 ?>
```



На листинге 7 отображён запрос на обновление информации в базе данных. Обновление происходит в таблице Ingredient, а в качестве поступающих на обновления данных являются название ингредиента, наименование изображения в формате png или jpeg (в качестве наименования изображения также указывается формат самого изображения, поэтому в базе данных изображение будет указываться как пример строки «beze.png»), а также указывается описание ингредиента. В качестве условия обновления указывается, что в столбце idIngredient будет совпадать с ID ингредиента, который был взят из глобального массива \$_GET [10]. В случае успешной отправки и обработки запроса выводится соответствующее сообщение. В случае наличия невыполнения условий выводится соответствующее сообщение с описанием ошибки.

Далее было необходимо реализовать удаление ингредиента из базы данных. Блок с удалением ингредиента реализован аналогично блоку с обновлением информации ингредиента. Поиск в удалении ингредиента имеет такое же код, как в листинге 5. Контрольный пример представлен на рисунке 5.

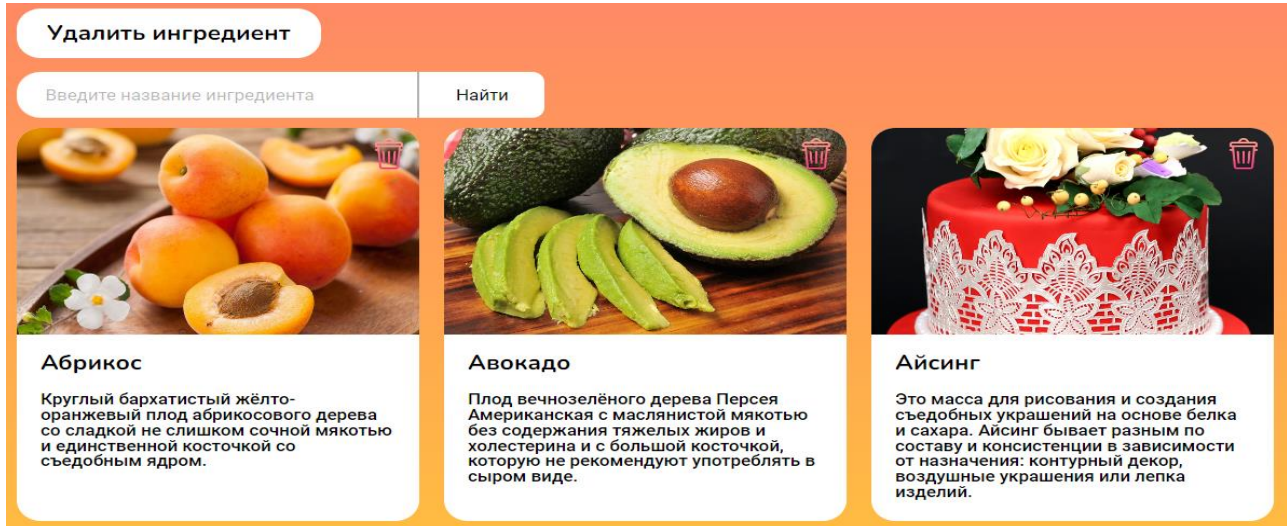


Рис. 5 - Контрольный пример вывода ингредиентов при поиске
в блоке удаления ингредиентов

На контрольном примере отображён список ингредиентов, при вводе в поиск строки «А». В правом верхнем углу карточки каждого ингредиента отображена мусорная корзина, при нажатии на которую открывается форма для



отображения всей информации об ингредиенте из базы данных пользователю без возможности изменения информации. Реализация формы для удаления ингредиента представлена на листинге 8.

Листинг 8

```
update.php x
276 <form id="srdel" class="del-ingr-form" <?php if(isset($_GET['idd'])) {echo "show";} else {echo "hide";} ?>"
277   action="ingrdel.php" method="post" enctype="multipart/form-data">
278   <?php
279     if(isset($_GET['idd'])||isset($_SESSION['idIngredient'])) {
280       if(isset($_GET['idd']) && $_GET['idd']!=NULL) {
281         $_SESSION['idIngredient']=$_GET['idd'];
282         $deleteIngredient=$_SESSION['idIngredient'];
283         $sql="SELECT * FROM `Ingredient` WHERE `idIngredient`='{$_SESSION['idIngredient']}'";
284         $result=mysqli_query($connect,$sql);
285         $Ingredient=mysqli_fetch_assoc($result); }?>
286 <h3>Удаление ингредиента</h3>
287 <div class="img">
288   
290
291 <p>Название ингредиента: </p>
292 <input type="text" name="name" value="<?php if(isset($_GET['idd'])) { echo $Ingredient['name'];} ?>">
293
294 <p>Описание ингредиента: </p>
295 <textarea class="description-ingr" type="text" name="description" value="">
296   <?php if(isset($_GET['idd'])) {echo $Ingredient['description'];} ?></textarea>
297 <input type="submit" name="submit-ingrdel" value="Удалить">
298 </form>
```

При нажатии на кнопку в правом верхнем углу карточки ингредиента, ID выбранного ингредиента передаётся в глобальный массив `$_GET [10]`. Далее происходит отправка запроса на сервер для получения остальной информации об ингредиенте. Затем полученная информация отображается в форме без возможности редактирования. При нажатии на кнопку формы `input` с атрибутом `type="submit"` происходит отправка запроса на удаление из базы данных на сервер (листинг 9).


Листинг 9

```
ingrdel.php x update.php x
1 <?php
2 include ("config/connect.php");
3 session_start();
4 $sql="DELETE FROM `Ingredient` WHERE `idIngredient`='{$_SESSION['idIngredient']}'";
5 $result=mysqli_query($connect,$sql);
6 if($result) {
7   $_SESSION['ok']="Ингредиент успешно удалён";
8   header('Location: update.php');
9 } else {
10  $_SESSION['ok']="Удаление не прошло";
11  header('Location: update.php');
12 }
13
14 ?>
```



На листинге 9 отображена отправка запроса на удаление ингредиента из базы. Удаление происходит из таблицы Ingredient с условием, что ID ингредиента в базе совпадает с ID ингредиента в сессии. При успешном или безуспешном удалении ингредиента выводится соответствующее сообщение. Контрольный пример формы для удаления ингредиента представлен на рисунке 6.

Удаление ингредиента



Название ингредиента:

Авокадо

Описание ингредиента:

Плод вечнозелёного дерева Персея Американская с маслянистой мякотью без содержания тяжелых жиров и холестерина и с большой косточкой, которую не рекомендуют употреблять в сыром виде.

Удалить

Рис. 6 - Контрольный пример формы для удаления ингредиента

В данной работе была реализована информационная система для взаимодействия с ингредиентами в базе данных MySQL веб-сайта с рецептами. В качестве локального сервера было использовано такое программное обеспечение, как OpenServer, которое является бесплатным и лежит в открытом доступе. Был представлен фрагмент базы данных веб-сайта с рецептами, где идёт взаимодействие с таблицей Ingredient. Были реализованы основные запросы для взаимодействия с базой данных, как: отображение информации об



ингредиентах, добавление нового ингредиента в базу, обновление информации о существующем ингредиенте в базе, а также удаление информации о выбранном ингредиенте.

В качестве дальнейшего развития проекта предлагается добавление группы ингредиентам, такие как: кондитерские изделия, грибы, мука и мучные изделия, мясо и мясная продукция и т.д. Также возможность добавления групп ингредиентов, обновление информации о группах, удаление групп ингредиентов. В уже существующую систему может быть добавлена возможность изменения, добавления и удаления группы в самого ингредиента.

Данную систему можно использовать в существующих проектах, не связанных с пищевой тематикой, и добавлять новые возможности.

Список литературы:

1. Культура питания российского населения (по результатам социологического исследования) // Здоровье населения и среда обитания - ЗНИСО URL: <https://zniso.fcgie.ru/jour/article/view/818> (дата обращения: 10.05.2023).

2. Основная информация про MySQL // OpenNET URL: <https://www.opennet.ru/docs/RUS/mysqlcli/glava01.html> (дата обращения: 10.05.2023).

3. Что такое HTML: основные теги // skypro URL: <https://sky.pro/media/что-такое-html/> (дата обращения: 12.05.2023).

4. Основы CSS // mdn web docs URL: https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/CSS_basics (дата обращения: 14.05.2023).

5. Документация // OSPanel URL: <https://ospanel.io/docs/> (дата обращения: 11.05.2023).

6. Создание приложения 'База данных кулинарных блюд' // Библиофонд URL: <https://www.bibliofond.ru/view.aspx?id=871300> (дата обращения: 16.05.2023).



7. Управление сессиями // php URL: <https://www.php.net/manual/ru/book.session.php> (дата обращения: 17.05.2023).

8. База данных фото с помощью HTML, PHP и MySQL // Интернет технологии.ру URL: <https://www.internet-technologies.ru/articles/zagruzka-izobrazheniy-v-bazu-dannyh-ili-na-server-s-pomoschyu-html-php-i-mysql.html?ysclid=lhvqv4qxgk855034675> (дата обращения: 17.05.2023).

9. Как сделать поиск по базе данных в PHP // ProgNote.ru URL: <https://prognote.ru/web-dev/back-end/how-to-do-a-database-search-in-php/?ysclid=lhvqwrwj48823337436> (дата обращения: 18.05.2023).

10. \$_GET // php URL: <https://www.php.net/manual/ru/reserved.variables.get.php> (дата обращения: 18.05.2023).