

## СОДЕРЖАНИЕ

|   |    |
|---|----|
| Введение . . . . .                          | 2  |
| 1. Обзор трёхзвенной структуры . . . . .    | 3  |
| 2. Актуальность разработки сайтов . . . . . | 6  |
| 3. Структура сайта . . . . .                | 8  |
| 3.1. Front-end сайта . . . . .              | 8  |
| 3.2. Клиент-сервер . . . . .                | 12 |
| 4. Структура базы данных . . . . .          | 13 |
| 5. Код сайта . . . . .                      | 15 |
| 5.1. Главная страница . . . . .             | 15 |
| 5.2. Процедура выдачи подарков . . . . .    | 17 |
| Заключение . . . . .                        | 19 |
| Список использованных источников . . . . .  | 20 |

## **ВВЕДЕНИЕ**

На сегодняшний день многие дети перестали верить в Деда Мороза. Компьютерные игры и социальные сети отнимают все время молодых людей, делая их зависимыми от компьютера. Зависимость развилась настолько сильно, что дети разучились читать книжки, а тем более писать вручную. Специально для таких детей разработан сайт Дедушки Мороза ФКТиПМ. Чтобы отправить ему письмо, совершенно не нужно быть грамотным. Единственное требование к пользователю - умение читать и кликать мышью. Кроме того, ответ от Деда Мороза приходит мгновенно, не требуя ждать пользователя пару месяцев как в случае бумажных писем. Дедушка Мороз ФКТ и ПМ подберёт для каждого ребенка тот подарок, которого он заслуживает.

Целью работы является изучение связки php-html-css-javascript и увеличение новогоднего настроения среди населения.

World Wide Web – глобальная компьютерная сеть, на сегодняшний день содержит миллионы сайтов, на которых размещена всевозможная информация. Люди получают доступ к этой информации посредством использования технологий Internet. Для поиска по интернету используют специальные программы – Web-браузеры, которые существенно облегчают путешествие по бескрайним просторам интернета

## 1 Обзор трёхзвенной структуры

Как правило компьютеры и программы, входящие в состав информационной системы, не являются равноправными. Некоторые из них владеют ресурсами (файловая система, процессор, принтер, база данных и т.д.), другие имеют возможность обращаться к этим ресурсам. Компьютер (или программу), управляющий ресурсом, называют сервером этого ресурса (файл-сервер, сервер базы данных, вычислительный сервер). Клиент и сервер какого-либо ресурса могут находиться как на одном компьютере, так и на различных компьютерах, связанных сетью.

В рамках многоуровневого представления вычислительных систем можно выделить три группы функций, ориентированных на решение различных подзадач:

- функции ввода и отображения данных (обеспечивают взаимодействие с пользователем);
- прикладные функции, характерные для данной предметной области;
- функции управления ресурсами (файловой системой, базой данных и т.д.).

Выполнение этих функций в основном обеспечивается программными средствами, которые можно представить в виде взаимосвязанных компонентов, где:

- компонент представления отвечает за пользовательский интерфейс;
- прикладной компонент реализует алгоритм решения конкретной задачи;
- компонент управления ресурсом обеспечивает доступ к необходимым ресурсам.

Автономная система (компьютер, не подключенный к сети) представляет все эти компоненты как на различных уровнях (ОС, служебное ПО и утилиты, прикладное ПО), так и на уровне приложений (не характерно для современных программ). Так же и сеть — она представляет все эти компоненты, но, в общем случае, распределенные между узлами. Задача сводится к обеспечению сетевого взаимодействия между этими компонентами.

Архитектура «клиент-сервер» определяет общие принципы организации взаимодействия в сети, где имеются серверы, узлы-поставщики некоторых специфичных функций (сервисов) и клиенты, потребители этих функций.

Практические реализации такой архитектуры называются клиент-серверными технологиями. Каждая технология определяет собственные или использует имеющиеся правила взаимодействия между клиентом и сервером, которые называются протоколом обмена (протоколом взаимодействия).

Тенденция в клиент-серверных технологиях связана со все большим использованием распределенных вычислений. Они реализуются на основе модели сервера приложений, где сетевое приложение разделено на две и более частей, каждая из которых может выполняться на отдельном компьютере. Выделенные части приложения взаимодействуют друг с другом, обмениваясь сообщениями в заранее согласованном формате. В этом случае двухзвенная клиент-серверная архитектура становится трехзвенной (three-tier, 3-tier).

Как правило, третьим звеном в трехзвенной архитектуре становится сервер приложений, т.е. компоненты распределяются следующим образом: -представление данных — на стороне клиента. -прикладной компонент —

на выделенном сервере приложений (как вариант, выполняющем функции промежуточного ПО). -управление ресурсами — на сервере БД, который и представляет запрашиваемые данные.

О преимуществах трехзвенной структуры в сравнении с двухзвенной можно узнать [8], Трехзвенная архитектура может быть расширена до многозвенной (N-tier, Multi-tier) путем выделения дополнительных серверов, каждый из которых будет представлять собственные сервисы и пользоваться услугами прочих серверов разного уровня[1].

## **2 Актуальность разработки сайтов**

Несмотря на то, что интернет давно и прочно вошел в нашу жизнь, многие предприниматели и даже крупные фирмы не понимают, что им даст создание собственного сайта, ведь есть другие хорошо зарекомендовавшие себя проверенные способы саморекламы: телевидение, радио, СМИ, баннеры, флайеры и тому подобное.

У любой современной компании существует сайт. Это один из элементов престижа, ведь именно в Интернете потенциальные клиенты будут в первую очередь искать информацию о фирме. И если у нее нет хотя бы одностраничника с прайсом, это покажется подозрительным – насколько же это неуспешная фирма, если не может даже небольшой веб-ресурс создать?

Актуальность создания сайта состоит также в том, что если вы хотите донести информацию максимально быстро до огромного количества людей, то лучше, чем с помощью собственного сайта сделать это не получится никак. Веб-ресурс позволяет представить информацию о компании и ее товарах или услугах сжато и одновременно полноценно. Также сайт может сообщать о новостях фирмы, об изменениях в прайсе или режиме работы, содержать отзывы благодарных клиентов.

Совсем не каждая фирма нуждается в крупном портале со сложным дизайном и функционалом. Иногда бывает достаточно небольшого сайта-визитки, который можно сделать самостоятельно или же заказать профессионалам за небольшую плату.

Необходимо понимать, что ни один другой ресурс не даст столько преимуществ, сколько собственный сайт, будь это визитка, Интернет-магазин или любой другой веб-ресурс.

Нужно отметить, что одним только созданием сайта дело не ограничится. Его будет необходимо развивать и поддерживать, увеличивать конверсию и своевременно пополнять, а это задача не из легких. Однако все эти усилия и затраты сполна окупятся прибылями, которые принесет сайт.

Вы можете попробовать сделать сайт самостоятельно, но вскоре поймете, что поддержка и наполнение сайта требуют много времени, сил и специфических навыков. Доверьте свой веб-ресурс профессионалам Cetera Labs, и ваш сайт окажется в надежных руках команды профессионалов.

### **3 Структура сайта**

Сайт состоит из главной страницы и страницы ответа деда мороза. В первую очередь сайт делится на две равнозначные, но всё же разные части: back-end и front-end [2]. Средства для тех. реализации проекта (HTML, CSS, JS, PHP) были выбраны как наиболее распространённые и доступные. К тому же именно их нам необходимо изучить в рамках данного курса сетевых технологий.

#### **3.1 Front-end сайта**

Обе страницы сайда Деда Мороза ФКТиПМ объединены единым лаконичным стилем, который сразу увеличивает новогоднее настроение в разы. Страницы сайта написаны на языке разметки html [5] с использованием стилей css [6]. На главной странице (рисунок 3.1) можно видеть форму в виде письма. В эту форму необходимо ввести свои личные данные: имя, возраст и город, чтобы Дедушка мороз мог разыскать ребенка в случае, когда это необходимо . Ниже идет список из десяти пожеланий, которые ребенок хотел бы отправить Деду Морозу (рисунок 3.2). Желания хранятся в базе данных и подгружаются автоматически. Администратор сайта (Дед Мороз ФКТиПМ) может добавлять список возможных пожеланий через командную строку. Внизу находится строка с текстом пожелания и большая кнопка для отправки письма.



Официальный сайт Дедушки Мороза ФКТиПМ

Я

ФИО

Петр Владимирович Пупкин

Возраст

45

Из города

Нижний Новгород

- Мальчик
- Девочка

Желаю Дедушке Морозу

Рисунок 3.1 – главная страница сайта

Желаю Дедушке Морозу

|                                |
|--------------------------------|
| Здоровья                       |
| Счастья                        |
| Любви                          |
| Не хворай                      |
| Удачи                          |
| Чтоб в пробках не стоял        |
| Вкусных cookie                 |
| Зачета по веб программированию |
| Не встретить M&msов            |
| Комуналку поменьше             |
| Заккрыть ипотеку               |

И хочу сказать

С Новым Годом!

Рисунок 3.2 – список пожеланий для Дедушки Мороза

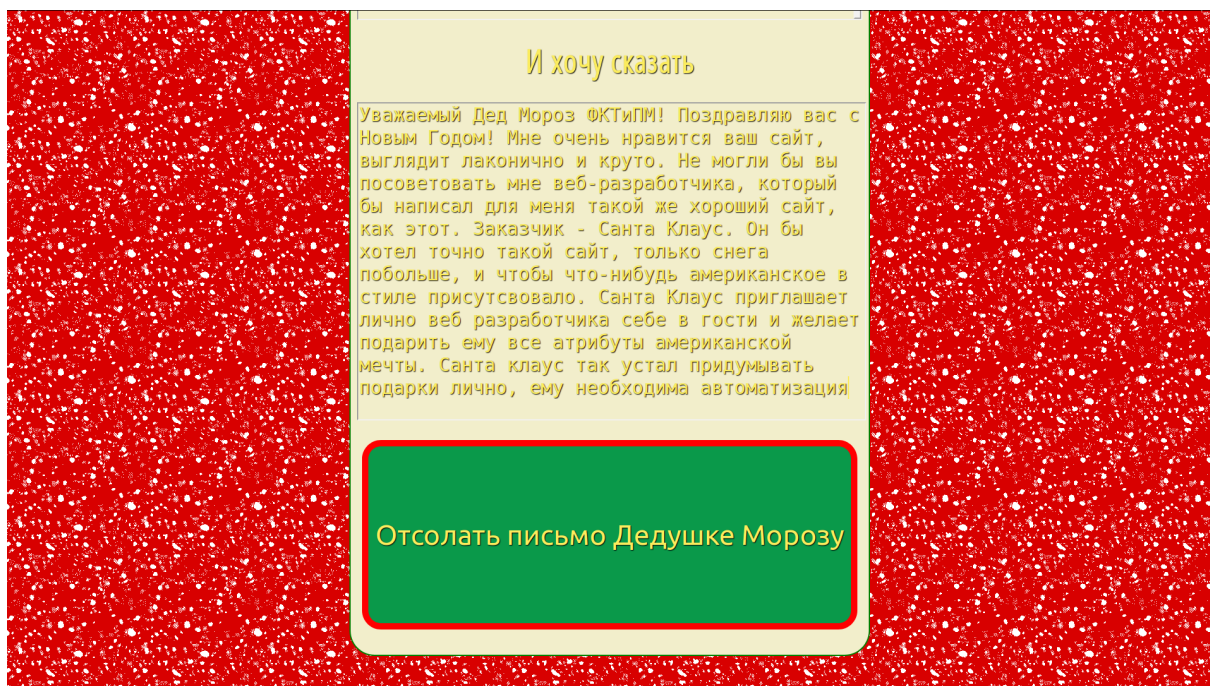


Рисунок 3.3 – строка ввода поздравлений и кнопка

На второй странице скрипт, написанный на javascript [7] выдает ответ в виде поздравлений и списка подарков. При этом вывод текста анимирован так, будто Дед Мороз сам пишет вам ответ(рисунок 3.4). Это реализовано с помощью бесплатной javascript библиотеки typed\_js [3].

Задачи зачётной работы:

- разработка сайта с использованием трехзвенной структуры;
- разработка дизайна оформления, делающего сайт более привлекательным для пользователей;
- создание удобного интерфейса для возможности комфортного пребывания пользователей на сайте;

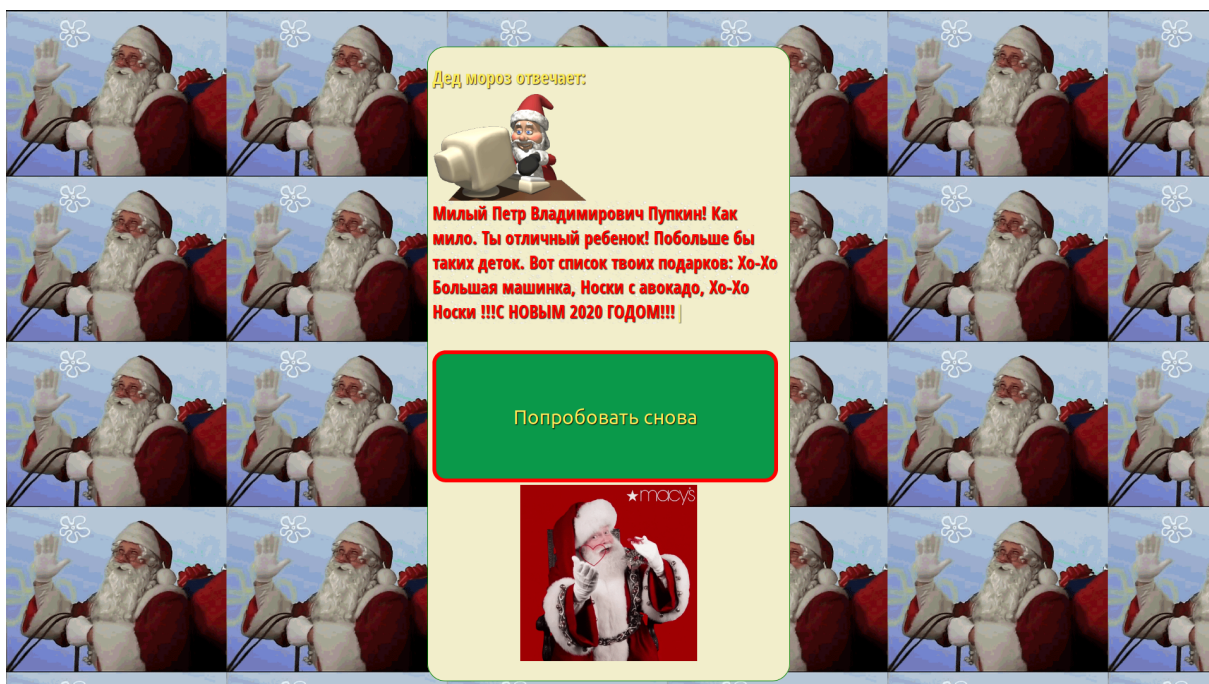


Рисунок 3.4 – строка ввода поздравлений и кнопка

На первой странице пользователь выбирает десять пожеланий Деду Морозу. Каждое желание имеет свой вес, причем этот вес может быть как положительным, так и отрицательным. Каждый подарок, в свою очередь имеет свою цену, причем цена подарка тоже может быть отрицательной. Каждый пользователь после заполнения первой страницы получает счёт, равный сумме весов желаний, которые они выбрали. Подарки от Деда Мороза выдаются по следующему алгоритму:

- 1) если счет отрицательный, рассматривать только отрицательные подарки;
- 2) если счет положительный, рассматривать только положительные подарки;
- 3) если список возможных подарков пустой или счет нулевой - конец;
- 4) из списка допустимых подарков выбрать случайный;
- 5) если модуль счет не меньше, чем модуль цены этого подарка:
  - а) добавить подарок в список подарков;

- б) уменьшить счет на цену подарка;
- 6) иначе удалить из списка возможных подарков;
- 7) перейти на шаг 3

В зависимости от счета пользователя, ему предлагается особенный стиль. Если, например, количество очков лежит в пределах от 0 до 20, то пользователю будет предложен стиль, в котором Дедушка Мороз его благодарит. Если же, например, количество очков меньше -50 - то пользователю предлагается агрессивный дизайн.

### **3.2 Клиент-сервер**

Клиент – сервер процесс описывает взаимодействие между двумя компьютерными программами, при котором одна программа (клиент) направляет запрос к другой программе (серверу), которая выполняет данный запрос. Как правило, несколько клиентских программ обращаются к одной общей серверной программе. Например, веб-браузер – это клиентская программа, обращающаяся с запросами (на отправку веб-страницы или файлов) к веб-серверу. В моей работе сервером является приложение Apache, в нем хранится СУБД, к которому обращается клиентская часть. Примером работы клиент-сервера является выдача подарков в соответствии с пожеланиями ребенка.

Back-end сайта реализован на языке php [4], используя СУБД Mysql. Веб сервером выступает популярный для php приложений Apache. В целом в реализации сайта Деда Мороза был использован стек технологий под названием LAMP (Linux, Apache, Mysql, PHP) Для реализации функционала сайта спроектирована база данных структуры, изображенной на 4.1.

## 4 Структура базы данных

База данных реализована с помощью СУБД Mysql, её структуру можно наблюдать на рисунке 4.1.

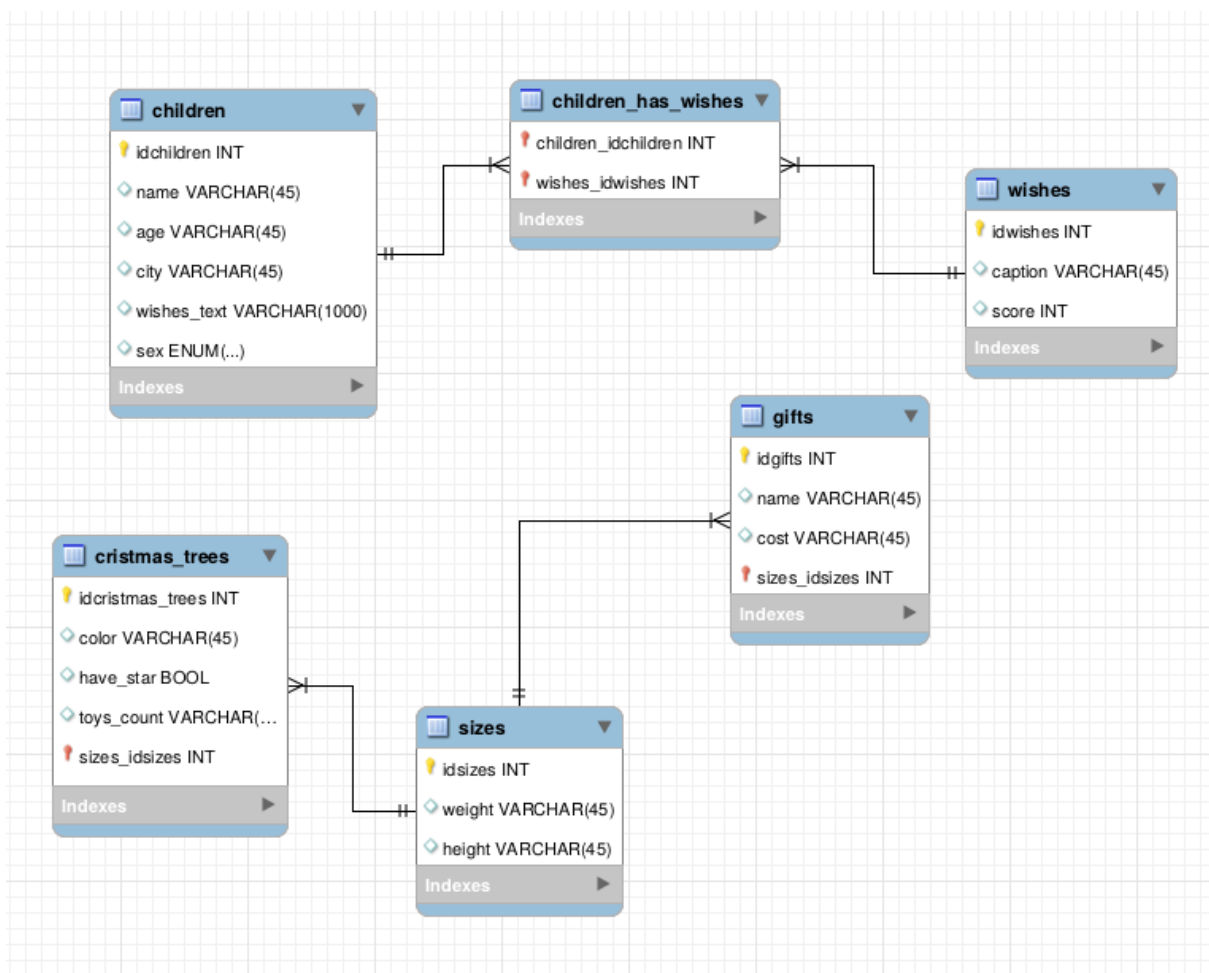


Рисунок 4.1 – структура базы данных

На стороне сервера реализованы две процедуры обращения к базе данных:

- 1) вставка в базу данных информации о ребенке;
- 2) выбока пожеланий текущего ребенка, счет количества его очков и выборка подарков по алгоритму, описанному выше.

Можно видеть пример запроса к базе данных из php на рисунке 4.2

```
1 <?php
2
3 if ($_SERVER["REQUEST_METHOD"] == "POST") {
4     $conn = new mysqli();
5     $sql = "select idchildren, name, caption, score from
6     children JOIN children_has_wishes ON
7     idchildren = children_idchildren
8     JOIN wishes ON wishes_idwishes = idwishes where idchildren = " . $id;
9
10    // Получили список подарков
11    $result = $conn->query($sql);
12
13    //Посчитали кол-во очков
14    $sum = 0;
15    $score_name = "score";
16    while ($row = $result->fetch_array()){
17        $sum += (int)$row[$score_name];
18    }
19
20    function compute_cost(int $inp){
21        return $inp;
22    }
23
24    function most_exp($cash,$myskli_res){
25        $row = $myskli_res->fetch_array();
26        $cash dif = abs($cash - $row["cost"]);
27        $result_row = $row;
28
29        while ($row = $myskli_res->fetch_array()){
30            $local_cashdif = abs($cash - $row["cost"]);
31            if ($local_cashdif < $cash dif){
32                $cash dif = $local_cashdif;
33                $result_row = $row;
34            }
35        }
36
37        $myskli_res->data_seek(0);
38        return $result_row;
39    }
40
41 }
42
43 }
```

Рисунок 4.2 – пример запроса к базе данных

## 5 Код сайта

Код сайта весьма сложен и длинен, поэтому в рамках отчета, его не удастся рассмотреть во всей полноте, однако можно привести важный составляющее его.

### 5.1 Главная страница

```
<!DOCTYPE html>

<html>

<html>

<head>

<meta charset="utf-8">

<link href="https://fonts.googleapis.com/css?family=
Open+Sans+Condensed:300" rel="stylesheet">
<link href="styles/style.css" rel="stylesheet" type="text/css">
<title>

</title>

</head>

<header>

<div class=site_header>

</div>

</header>

<body>
```



```

<form action="gift.php" method="post">
<h1>                !</h1>
<input type="hidden" name="example" value="data1">
<h2>        </h2>
<input name="name" type="text" required><br>
<h2>                </h2>
<input name="age" type="number" min="0" required><br>
<h2>        </h2>
<input name="city" type="text" required><br>
<h2>
        <input type="radio" name="gender" value="male" checked>
                <br>
        <input type="radio" name="gender" value="female">        <br>
</h2>

<h1>        ,                :</h1>

<?php include 'include/wishes_select.php';?>
<h1>        </h1>
<textarea name="message" rows="1">                !</textarea>
<br>
<h1>
<input type="submit" value="                "/>
</h1>
</form>
</body>
</html>

```



## 5.2 Процедура выдачи подарков

```
// push only gift with equal sign
while ($row = $result->fetch_array()){
    if ( sign((int)$row['cost']) == sign($sum)){
        $gift_list->push(array((int)$row['cost'],$row['name']));
    }
}

$result_vect = new \Ds\Vector();
while ( ! $gift_list->isEmpty() and $cash != 0 ){
    // Choose random index
    $rand_gift = rand(0,$gift_list->count()-1);
    // If enough money
    if ( $cash - abs($gift_list[$rand_gift][0]) >= 0){
        // Gift it
        $result_vect->push($gift_list[$rand_gift][1]);
        $cash -= abs($gift_list[$rand_gift][0]);
    }
    else{
        // Remove from gift list
        $gift_list->remove($rand_gift);
    }
}

$gifts = $gifts . "                : ";
```

```

// Insert xoxo randomly in text
$xoxo = " - ";
$xoxo_count = rand(3,10);
$xoxo_vector = new \Ds\Vector();
for ($i=0; $i <= $xoxo_count; $i++) {
    $xoxo_vector->push(rand(0,$result_vect->count()-1));
}

for ($i=0; $i < $result_vect->count(); $i++) {
    //echo "string";
    while( $xoxo_vector->find($i) ){
        $gifts = $gifts . $xoxo;
        $xoxo_vector->remove($xoxo_vector->find($i));
    }

    $gifts = $gifts . $result_vect[$i] . ", ";

}

$gifts = quotes_js(rtrim($gifts,", "));

```

## **ЗАКЛЮЧЕНИЕ**

В рамках зачётной работы были рассмотрены основные методы проектировки и написания сайтов сети Интернет. В результате для рассматриваемых правил был разработан сайт на основе трёх звеньев (клиент – сервер – база данных) Деда Мороза ФКТиПМ. Сайт записывает информацию о всех пользователях, поддерживает интерактивное общение с пользователями благодаря библиотеке `typed.js`.

Были изучены основные технологии создания веб приложений - серверный язык `php`, язык разметки `css` и браузерный язык `javascript`. Кроме того, изучены основы администрирования сайтов на примере популярного веб сервера `Apache` на платформе `Linux`. Рассмотрены возможности использования баз данных на примере СУБД `Mysql`.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Архитектура «клиент-сервер» | Сетевые технологии / (Рус.). – URL: <http://www.4stud.info/networking/lecture5.html> [25 мая 2019]
- 2 Синица, С. Г. Веб-программирование и веб-сервисы : учебное пособие / С. Г. Синица ; М-во образования и науки Рос. Федерации, Кубанский гос. Ун-т. - Краснодар : [Кубанский государственный университет], 2013. - 158 с.
- 3 Библиотека анимации текста type\_js / (Eng) – URL: <https://mattboldt.com/demos/typed-js> [25 мая 2019]
- 4 Онлайн учебник по изучению серверного языка программирования php / (Eng) – URL: <https://www.w3schools.com/php/> [25 мая 2019]
- 5 Онлайн учебник по изучению языка разметки html / (Eng) – URL: <https://www.w3schools.com/html/> [25 мая 2019]
- 6 Онлайн учебник по изучению языка стилей css / (Eng) – URL: <https://www.w3schools.com/css/> [25 мая 2019]
- 7 Онлайн учебник по изучению браузерного языка javascript / (Eng) – URL: <https://www.w3schools.com/js/> [25 мая 2019]
- 8 Почему не стоит использовать двухуровневую архитектуру при разработке клиент-серверных приложений / (Ru) – URL: <https://habr.com/ru/post/348946/> [25 мая 2019]