МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВПО «КубГУ»)**

**Кафедра вычислительных технологий**

**КУРСОВАЯ РАБОТА**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Степаненко К.А. (подпись, дата)

Факультет компьютерных технологий и прикладной математики, курс 3

Направление 010300.62 – «Фундаментальная информатика и информационные технологии»

Научный руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Приходько Т.А. (подпись, дата)

Нормоконтролер \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Пашенцева В.В.

(подпись, дата)

Краснодар 2015

**СОДЕРЖАНИЕ**

1. ВВЕДЕНИЕ.............................................................................................. 3

1.1 Обзор аналогичных решений ....................................................... 4

1.2 Минусы, выявленные в результате анализа уже готовых решений................................................................................................ 6

1.3 Концепция разработки нового чата.............................................. 8

2. ПОДРОБНОЕ ОПИСАНИЕ ВСЕХ ФУНКЦИЙ ЧАТА И ТЕХНОЛОГИЙ, С ПОМОЩЬЮ КОТОРЫХ ОНИ РЕАЛИЗОВАНЫ………………...……………………… 9

3. РЕЗУЛЬТАТЫ РАБОТЫ………………………………………………………………………. 27

4. ЗАКЛЮЧЕНИЕ…………………………………………………………………………………… 30

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ……………………………………………. 31

**ВВЕДЕНИЕ**

С каждым годом ритм жизни всё выше и выше, и на живое общение остается не так много времени, как хотелось бы, но потребность людей в общении всегда будет актуальна. С давних пор человечество придумало массу устройств для передачи и приёма информации, как в одностороннем порядке, так и в двустороннем. В наше время, на помощь людям приходят информационные технологии, которые развиваются всё быстрее и быстрее. На смену пейджерам и телеграфам, которые долгое время использовались для передачи информации, пришла глобальная компьютерная сеть Интернет, которая расширяет возможности общения в режиме реального времени, почти в любом месте пребывания абонента.

Сейчас существует множество средств общения: ICQ, XMPP (ранее известный как Jabber), WhatsApp, Telegram, форумы, различные социальные сети и, конечно же, чаты. Целью работы, всех перечисленных средств общения, является как можно более близкая имитация обычного разговора.

**1.1 ОБЗОР АНАЛОГИЧНЫХ РЕШЕНИЙ**

Прежде чем приступить к работе, автором был сделан обзор существующих средств общения. В результате анализа стало ясно, что их можно поделить на две категории:

1. Средства общения в реальном времени (синхронные).
2. Асинхронные средства общения.

К синхронным средствам коммуникации относятся те, которые позволяют собеседникам видеть и слышать друг друга в реальном времени (например, Skype). Под реальным временем так же понимается процесс обмена информацией, при котором у общающихся имеется возможность получать ответные сообщения с минимальной задержкой. К этому типу относятся и чаты.

Чат, чаттер (от англ. chatter — болтать) — средство обмена сообщениями по компьютерной сети в режиме реального времени, а также программное обеспечение, позволяющее организовывать такое общение. Существует несколько видов программной реализации чатов:

1. HTTP или веб-чаты. Подобные чаты широко распространились еще в 90-х годах, но остаются популярными до сих пор. Данные чаты представляют собой обычную веб-страницу, на которой можно прочесть последние сообщения, написанные участниками чата. Страница чата постоянно обновляется с заданной периодичностью. Популярность таких чатов постепенно падает, связанно это с появлением новых технологий, которые позволяют подгружать новые сообщения без перезагрузки страницы.
2. Чаты на основе Flash или AJAX. В отличие от HTTP чатов, в этом случае, вместо периодической перезагрузки страницы, между клиентом и сервером открывается сокет, что позволяет моментально отправлять или получать сообщения. Еще одно отличие от HTTP чатов – меньшее количество расходуемого трафика (для медленного интернет соединения это является важной особенностью). К слову, в данной курсовой работе я буду делать чат на основе AJAX и PHP. [1, 2, 3]
3. Протокол IRC. Данный протокол был создан в 1988 году и служит для обмена сообщениями в режиме реального времени. Разработан в основном для группового общения, что позволяет общаться через личные сообщения и обмениваться данными, в том числе и файлами.

К асинхронным средствам общения относятся гостевые книги, веб-форумы и блоги.

Гостевые книги представляют собой упрощенный вариант веб-форума. Применяются на сайтах и позволяют посетителям сайта оставлять пожелания или отзывы, которые адресованы администратору или другим посетителям. С технической точки зрения, гостевая книга представляет собой программу на языке программирования PHP, но она может быть написана и с помощью HTML. Гостевая книга – распространенный способ обратной связи в Интернете.

Веб-форум – класс веб-приложений для организации общения посетителей сайта. Суть работы форума заключается в создании пользователями своих тем с последующим общением внутри этих тем. Форум отличается от чата тем, что позволяет общаться не в реальном времени. Это располагает к более серьёзным обсуждениям, поскольку отвечающему предоставляется больше времени на обдумывание ответа. Форумы часто используются в работе служб технической поддержки.

Блог — веб-сайт, основное содержимое, которого, регулярно добавляемые записи, содержащие текст, изображения или мультимедиа. Для блогов характерны недлинные записи временной значимости, упорядоченные в обратном хронологическом порядке (последняя запись сверху). Отличия блога от традиционного [дневника](https://ru.wikipedia.org/wiki/%D0%94%D0%BD%D0%B5%D0%B2%D0%BD%D0%B8%D0%BA) обусловливаются средой: блоги обычно публичны и предполагают сторонних читателей.

**1.2 МИНУСЫ, ВЫЯВЛЕННЫЕ В РЕЗУЛЬТАТЕ АНАЛИЗА УЖЕ ГОТОВЫХ РЕШЕНИЙ**

Перед тем, как приступить к работе, нужно было выявить основные преимущества и недостатки уже готовых и используемых решений. Для проведения такого анализа было взято три наиболее популярных чата [1, 2, 3].

После анализа большого набор чатов, было выявлено множество недостатков, которые портят впечатление о чате уже после нескольких часов проведенных в нём:

Первый минус. Отсутствие регистрации. Для того, чтобы получить доступ к рассмотренным чатам необходимо было лишь ввести логин, таким образом заблокированный пользователь может через несколько секунд получить доступ к чату уже под другим логином.

Второй минус. Огромное количество рекламы. Практически во всех чатах, реклама занимает примерно 10% монитора, из-за этого находиться в таких чатах весьма некомфортно. В добавок, во многих чатах отсутствует фильтр рекламы, поэтому можно было увидеть даже ту рекламу, которая никак не соответствует интересам его участника.

Третий минус. Действия в новых вкладках. В некоторых чатах, при переходе по пунктам меню, открываются новые вкладки. Таким образом, после двух-трёх переходов по сайту, будет открыто минимум три или четыре новые вкладки. Такой процесс не нравится пользователям, и они уходят с таких чатов, об этом можно говорить, если взглянуть на статистику посещения такого чата и количества оставленных там сообщений.

Четвертый минус. Отсутствия списка пользователей, которые онлайн. При входе в чат, не видно, кто сейчас в сети и кому можно написать. Во многих чатах есть только информация о том, сколько человек находится в чате. Из-за того, что их логины скрыты, очень сложно обращаться к какому-либо пользователю, так как его логин приходится писать вручную.

Пятый минус. Отсутствие выбора темы общения. Во всех рассмотренных чатах отсутствовала возможность выбора темы для общения, т.е. все сообщения отображались в одном общем окне.

Проведя анализ трёх популярных чатов, было выявлено пять проблем, которые заметно ухудшают юзабилити и общее впечатление от чата. Но следует также учитывать, тот факт, что одна проблема может порождать другую.

**1.3 КОНЦЕПЦИЯ РАЗРАБОТКИ НОВОГО ЧАТА**

Проанализировав готовые решения и выявив очевидные минусы, было принято решение написать собственный чат. Проектируемый программный продукт рассчитан на мгновенную доставку сообщений всем пользователям, находящимся в пределах одной ""комнаты"" ("комнатой" будем называть обсуждение одной тематики). За счет простого интерфейса, JavaScript’ов и простых функций на PHP, ожидается, что разрабатываемый чат будет работать очень быстро и часть кода будет исполняться сразу же в браузере пользователя, а это означает меньшую нагрузку на сервер. Так же благодаря технологии Ajax, которая заключается в «фоновом» обмене данными браузера с веб-сервером, некоторые страницы чата, при обновлении данных, не перезагружаются полностью, что сделает чат более удобным и быстрым. Все пользовательские данные планируется хранить в базе данных, а работа с ней будет происходит с помощью свободной реляционной системы управления базами данных - MySQL, которая разработана и поддерживается компанией Oracle.

Чат проектируется так, чтобы быть избавленным от тех недостатков, которые были выявлены в разделе 1.2, а также будет дополнен новыми функциями, которые не были найдены в других чатах, но которые имеют большое значение для комфортного общения и времяпровождения.

**2. ПОДРОБНОЕ ОПИСАНИЕ ВСЕХ ФУНКЦИЙ ЧАТА И ТЕХНОЛОГИЙ, С ПОМОЩЬЮ КОТОРЫХ ОНИ РЕАЛИЗОВАНЫ**.

Регистрация.

Во многих существующих чатах нет системы регистрации пользователей, что делает чат менее привлекательным и ограниченным по своему содержанию. Отсутствие регистрации влечет за собой отсутствие системы рейтинга, анкет пользователей и множества других возможных функций чата, которые бы привлекали пользователей. Система регистрации – это первое с чем столкнется пользователь на сайте, таким образом она планируется максимально упрощенной, чтобы не испортить первое впечатление пользователя о чате.

При разработке регистрации, нужно учитывать следующие пункты, которые являются базовыми и используются практически на всех сайтах, где присутствует система регистрации:

* длина логина должна быть не менее трёх символов и не более 20 символов
* логин должен состоять только из букв латинского алфавита и цифр
* длина пароля не должна быть менее 6 символов
* логин должен быть свободен

В чате окно регистрации содержит четыре поля: логин, пароль, повторение пароля, адрес электронной почты.

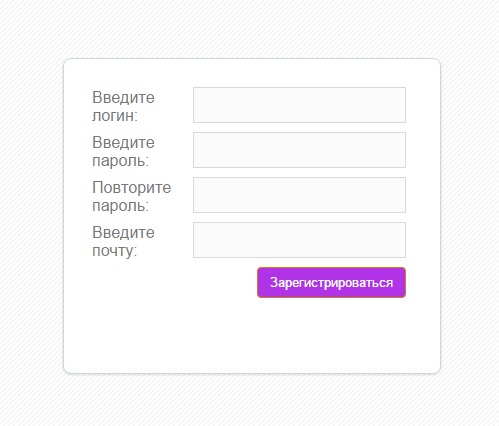


Рисунок 1 – Форма регистрации

Для проверки правильности ввода данных используется библиотека JQuery, которая фокусируется на взаимодействии Javascript’а и HTML. Данная библиотека позволяет оповещать пользователя о некоторых ошибках еще до отправки данных на сервер, таким образом уменьшается и нагрузка на сервер, так как данные проверяются непосредственно в браузере пользователя, а не на сервере. Для контроля введенных данных используются регулярные выражения, с помощью которых можно задать определенный формат нужного поля. Например, правильность ввода адреса электронной почты проверяется с помощью функции is\_email, в которой используется регулярное выражение:

function is\_email($email)

return preg\_match("/^([a-zA-Z0-9])+([\.a-zA-Z0-9\_-])\*@([a-zA-Z0-9\_-])+(\.[a-zA-Z0-9\_-]+)\*\.([a-zA-Z]{2,6})$/", $email);

Если все данные введены корректно, то они отправляются на сервер и уже там проходит новая обработка, а именно проверяется свободен ли логин. После проверки пользователь может получить от сервера два возможных варианта ответа:

1. Логин занят. В данном случае пользователю нужно ввести новый логин.
2. Пользователь успешно зарегистрирован. В данном случае данные пользователя успешно записаны в базу данных.

Авторизация

Перед тем, как начать общаться в чате, первое что ему нужно сделать – пройти процесс авторизации (при условии, что данный пользователь зарегистрировался ранее). Авторизоваться можно двумя способами: ввести логин и пароль или пройти авторизацию с помощью одной из социальных сетей. Для того, чтобы пользоваться возможностью авторизации через социальные сети, был подключен специальный виджет uLogin [4], который распространяется в свободном доступе. Благодаря такому виджету, пользователю не нужно вводить логин и пароль, а авторизация в чате занимает не более трёх секунд. После выбора авторизации и выполнения всех необходимых действий, данные пользователя отправляются на сервер, где после обработки может быть получено три возможных ответа:

1. Авторизация пройдена успешна.

После успешной авторизации необходимо запомнить, что пользователь авторизован. Проанализировав возможные варианты хранения информации об авторизации, я нашёл два наиболее часто встречающихся способа: coockie (куки) и session (сессии). Для выявления наиболее подходящего для текущей разработки способа, было принято решение проанализировать плюсы и минусы каждого способа.

Coockie – это файл, который хранится у пользователя в браузере и данные из этого файла передаются серверу в каждом запросе.

Данные таким способом передаются как текст, а значит – их легко перехватить. Так же к минусам относится и тот факт, что если мы решим хранить в куках логин пользователя “User1”, то он может открыть управление куками в своём браузере, найти нужную запись и исправить ее на, например, “Admin”. Таким образом, мы будем получать запись coockie, которая будет сообщать, что логин пользователя “User1” это “Admin”. Так же coockie можно украсть, а значит получить логин и пароль. Если украденный пароль зашифрован только md5, то его можно расшифровать с помощью специальных баз, которые доступны для загрузки в Интернете.

Session(сессии) используются для хранения временных данных при переходах между страницами одного сайта. При использовании сессий данные сохраняются во временных файлах на сервере. У сессии существует естественный срок годности - до закрытия браузера. Если закрыть только страницу, то при открытии сайта снова, данные о пользователе всё равно будут доступны. Более того, ввиду того, что каждая сессия имеет уникальный идентификатор, то смысл воровства идентификатора сессии (а больше ничего взять не получится) уже отсутствует. Но именно по причине временного хранения сессии, появляется большой минус сессий - неудобство. При закрытии браузера, сессии удаляются, а значит нужно проходить процедуру авторизации заново, что не совсем удобно, но более безопасно.

Таким образом, выявив плюсы и минусы coockie и session, было решено остановиться на сессиях, так как считается, что безопасность – самое главное, что должно быть на сайте.

При авторизации происходит подключение к базе данных, в которой происходит поиск введённого логина и сравнение паролей.

$query = mysqli\_query($connection, "SELECT `ban`,`why\_ban`,`prior`, `firstname`, `surname` FROM `user` WHERE `login` = '$login' AND `password` = '$password'");

$user\_info = mysqli\_fetch\_array($query);

Пароли в базе хранятся в зашифрованном виде. Шифрование происходит с помощью 128-битного алгоритма шифрования(MD5). Поэтому прежде чем делать запрос в базу данных, введенный пароль шифруется.

$password = md5(md5($password))

Например, после такого шифрования, пароль 12345 будет иметь вид:

1f32aa4c9a1d2ea010adcf2348166a04

Если логин найден, а пароли совпадают, то происходит запись информации в сессии.

$\_SESSION['login'] = $login;

$\_SESSION['prior'] = $id\_user['prior'];

1. Авторизация не пройдена.

Причинами может быть не правильный логин, пароль или пользователь уже находится в сети. При не удачной авторизации пользователю выводится сообщение об ошибке и предлагается пройти авторизацию еще раз.

1. Аккаунт заблокирован.

Может быть и такое, что аккаунт пользователя заблокирован, причин на это может быть много, но основная – нарушение правил чата. Данные о блокировке аккаунта хранятся в базе данных и извлекаются при авторизации. Заблокировать пользователя может либо модератор, либо администратор. При блокировке обязательно указывается причина и логин того, кто заблокирован. Подробнее механизм блокировки будет изложен далее.

**Действие при первом входе после регистрации**

Чат не предназначен для анонимных пользователей, поэтому при первом входе после регистрации, пользователю предлагается заполнить некоторые данные о себе (имя и фамилия). Только после заполнения этих данных пользователю будет открыт доступ к главной странице сайта.

**Главная страница**

Если регистрация пройдена успешно и необходимые данные внесены в базу, то пользователь попадет на главную страницу. Здесь пользователю видны почти все доступные функции чаты. Визуально главную страницу можно разделать на 3 блока:

левый блок – меню;

центральный блок – основная информация;

правый блок – дополнительные функции;

В меню выводятся все доступные функции для данного пользователя. Количество доступных функции зависит от рейтинга пользователя и его должности в чате.

В блоке основная информация изначально выводятся новости чата. Далее информация в этом блоке будет меняться.

Роль дополнительных функций на главной странице выполняет список доступных комнат. Данный блок реализован с помощью JQuery, что позволяет загружать список доступных комнат без перезагрузки страницы. Таким образом, при выборе определенной категории, пользователю будет предложен список доступных комнат, который будет отображен в блоке основной информации.

Получение списка комнат происходит из базы данных, в которой хранится id "комнаты", ее название, количество максимально возможных пользователей для этой "комнаты" и пароль "комнаты". Обновление "комнат" в блоке основной информации доступно раз в 20 секунд, это сделано с целью уменьшения запросов к базе данных, а значит и уменьшения вероятности перегрузки базы данных.

**Вход в комнату**

В чате существует два типа комнат: с паролем, без пароля.

Рассмотрим "комнаты" с паролем.

Пароль устанавливается создателем "комнаты" и нужен для того, чтобы ограничить доступ к комнате не знакомым пользователям, т.е. это сделано для того, чтобы общаться только с отдельными пользователями. При входе в такую комнату первым делом пользователю нужно ввести пароль "комнаты". При вводе пароля происходит соединение с базой данной и если пароли совпадают, то пользователю разрешается доступ в комнату, а в переменную сессии устанавливается значение равное id "комнаты". Установка такого значения нужна для того, чтобы пользователь, войдя в комнату, не имел доступа к другим комнатам (т.е. ему разрешено находиться только в одной "комнате") и если он попытается перейти в другую "комнату" “не легальным” способом, то он моментально будет возвращен в ту комнату, к которой подключился ранее.

**"Комнаты" без пароля**

Доступ к таким комнатам разрешается любым пользователем, т.е. при входе в чат пользователю сразу доступно окно чата. При доступе к таким комнатам пользователю так же устанавливается специальный флаг, который запрещает переход в другие "комнаты", если пользователь находится уже в какой-либо комнате.

**Окно чата**

При входе в комнату, пользователю доступно в левом блоке меню, в который добавляется кнопка “Выйти из "комнаты"”.

В поле основной информации выводится тема "комнаты", сообщения чата, поле ввода сообщений, кнопка отправки сообщений и панель выбора смайликов.

Я расскажу о каждом пункте поля основной информации в комнате, а также о технологиях, которые были применены для реализации каждого пункта.

**Тема "комнаты"**

Для каждой "комнаты" доступно краткое описание того, о чем идёт разговор в комнате. Таким образом, вошедшему в комнату пользователю сразу будет известно о чём идёт разговор в данном чате, а значит пользователь может сразу определиться интересна ли ему эта тема или нет, что способствует экономии времени и более быстрому выбору подходящей "комнаты". Тема каждой "комнаты" хранится в общем XML файле, что уменьшает нагрузку на базу данных.

2. Раздел сообщений.

Все сообщения чата хранятся в txt файле, что значительно повышает работу чата. Для каждой "комнаты" создается отдельный txt файл, который имеет вид

logID.txt, где ID -–id "комнаты"

В данном файле хранятся последние 25 сообщений чата. Сделано это для того, чтобы уменьшить размер хранимых данных на сервере. При отправке нового сообщения происходит проверка количества сообщений в окне чата, если их более 25, то удаляется самое ранее сообщение (т.е. самое первое которое находится в окне чата).

Обновление сообщений происходит каждую секунду, пользователь не замечает этой задержки, но она важна чтобы уменьшить нагрузку на сервер. Обновление происходит с помощью Javascript’a, а именно с помощью функций библиотеки JQuery, что позволяет загружать новые сообщения без перезагрузки окна браузера. За загрузку сообщений отвечает небольшой фрагмент кода:

function loadLog()

{

$.ajax ({

url: "log\_chat/log<?php **echo** $\_SESSION['id\_room'] ?>.txt",

cache: false,

success: function(text){

$("#chatbox").html(text);

},

});

}

Отправка сообщений в чат происходит также без перезагрузки страницы. Когда сообщение поступает на сервер, то первым делом оно обрабатывается. Обработка сообщения происходит в файле post.php. В данном файле сообщение проверяется на наличие смайликов, а затем преобразуются к специальному виду, т.е. к сообщению добавляется дата его отправки и имя отправителя. Только после обработки, сообщение добавляется в файл "комнаты".

Обработка смайликов происходит с помощью процедуры find\_smile(text), где text – исходное сообщение. Изначально смайлики представлены в текстовом виде, но после обработки данной процедурой, смайлики будут представлены в графическом виде.

Для того, чтобы пользователь мог точно знать, что его сообщение отправлено используется звуковой сигнал, который звучит только после того, как сообщение отправлено на сервер. По желанию звуковой сигнал можно отключить.

Когда все основные функции отправки и доставки сообщений были реализованы, то было принято решение произвести тестирование, чтобы убедиться, что все функции работают корректно и не возникает задержек с отправкой и доставкой сообщений. На этапе тестирования была выявлено, что можно отправлять в чат пустые сообщения. Изначально эта ошибка была устранена следующим образом: когда сообщение было отправлено на сервер, то в файле post.php, проверялась его длина, если она больше нуля, то сообщение проходило на дальнейшую обработку. В противном случае ничего не происходило, т.е. сообщение не добавлялось в чат. Казалось, что проблема устранена, но после очередного тестирования было замечено, что при отправке пустого сообщения срабатывает звуковой сигнал, а это значит, что сервер делает одну ненужную операцию – начальная обработка пустого сообщения. Данная проблема была решена с помощью Javascript. Отправленное сообщение в браузере пользователя проверялось на то, чтобы его длина была больше нуля и только после этого происходит отправка сообщения на сервер и срабатывание звукового сигнала. Таким образом удалось избавить сервер от выполнения лишней операции, а значит решить и ряд других небольших, но важных проблем.

Под полем, в которое набирается сообщение, находится панель смайликов.

Панель смайликов помогает пользователю быстро вставить в текст нужный смайлик. Данная панель разработана с помощью библиотеки JQuery, т.е. вставка смайлика в текст происходит моментально, без перезагрузки страницы.

В правом блоке окна доступна информация о тех, кто сейчас находится в данной комнате. Данные, о тех пользователях, которые сейчас в комнате находится в файле log\_onlineID.txt, где ID – номер соответствующей "комнаты". Данные о пользователе обновляются по тому же принципу, что и сообщения в чате, но в целях снижения потребляемого трафика, а также уменьшения нагрузки на сервер, данные обновляются каждые 5 секунд.

**Изменение настроек пользователя**

Каждый пользователь может настроить чат так, чтобы ему было комфортно проводить время на данном сайте. Настройки каждого пользователя хранятся в отдельной таблице базы данных, а при авторизации происходит загрузка этих настроек из базы данных и формирование страницы, которая соответствует настройкам пользователя. Сейчас я расскажу об основных настройках, который может сделать любой зарегистрированный пользователь.

1. **Выбор темы оформления.**

На сайте доступно две темы оформления: темная и светлая. При регистрации, по умолчанию ставится светлая тема, но её можно изменить в любой момент. Для того, чтобы менять тему оформления на сайте, есть несколько возможных решений:

Создание двух одинаковых страниц. Отличие этих страниц будет лишь в том, что на одной будет подключаться файл с светлой темой оформления, а на другой с темной. Данный метод – это не самый лучший вариант для выхода из данной ситуации, так как у каждой страницы будет свой “двойник”, который занимает место на сервере. Еще одним минусом данного метода является тот факт, что при редактировании какой-то области сайта, нужно будет проделывать одну операцию несколько раз. Пожалуй, главный минус этого метода – большое количество файлов.

Еще одно решение – создание переменной сессии. Данный метод наиболее подходящий в данной ситуации, так как для его применения не нужно создавать копию страницы. Суть этого метода в следующем: при авторизации, из базы данных считываются настройки. Затем проверяется тот параметр, который отвечает за тему оформления, если он указывает на то, что тема должна быть темной, то в переменную сессии заносится некоторый флаг:

$\_SESSION[‘themes’] = 1;

А если тема оформления светлая, то переменная сессии даже не создается. Таким образом при переходе на какую-либо страницу, будет производится проверка: если переменная сессии существует, то будет загружен файл с темной темой оформления, в противном случае будет загружен файл с светлой темой оформления.

**2. Смена пароля**

Смена пароля – стандартная функция для большинства сайтов. Окно, в котором происходят все действия состоит из трех полей: новый пароль, повтор нового пароля и старый пароль. Проанализировав множество чатов, я заметил, что у большей части чатов при смене пароля не требуется ввод старого пароля. Ввод старого пароля – одна из мер безопасности и гарантия того, что пароль вводит хозяин страницы, а не человек, который получил несанкционированный доступ к станице. При смене пароля в первую очередь делается проверка на то, что пароли в полях “новый пароль” и “повторите новый пароль” совпадают. Если пароли не совпадают, то пользователь видит перед собой соответствующую ошибку, в противном случае данные передаются на сервер, где происходит вызов функции changePassword($new\_pass, $old\_pass), которая находится в файле funtions.php. В данной процедуре происходит подключение к базе данных, затем сравнивается введенный старый пароль с тем, что хранится сейчас в базе данных. Если пароли совпадают, то старый пароль перезаписывается на новый, в противном случае выводится сообщение об ошибке.

changePassword($new\_pass, $old\_pass)

{

$login = $\_SESSION['login'];

$old\_pass = md5(md5($old\_pass));

$new\_pass = md5(md5($new\_pass));

$query = "UPDATE `user` SET password = '$new\_pass' WHERE `login` = '$login' AND `password` = '$old\_pass'";

$result = mysqli\_query($connection, $query);

}

**3. Смена имени и фамилии**

Как я уже отмечал ранее, чат не предназначен для анонимных пользователей, поэтому сменить имя и фамилию, введенные при регистрации является сложным процессом, который может быть отклонен администрацией. На странице смены пользователь увидит три поля для ввода: имя, фамилия, причина смены. После заполнения всех полей, данные отправляются на сервер, где помещаются в XML файл. Далее решение о смене имени и фамилии будет принято администрацией чата. Для этого у администратора есть специальная панель управления, о которой будет рассказано позже.

**4. Включение/отключение звуковых оповещений**

Благодаря данному пункту настройки, у пользователя есть возможность отключить звуковые сигналы при отправке и получении сообщений. Данные настройки так же сохраняются в базу данных и извлекаются при авторизации пользователя. При извлечении настроек из базы данных, создается переменная сессии в том случае, если звуковой сигнал включен пользователем, если же звуковой сигнал выключен, то процесс создания пропускается.

**5. Анкета пользователя**

У каждого пользователя есть своя анкета, данные которой хранятся в таблице info\_users. Анкета позволяет каждому пользователю указать некоторую информацию о себе. В анкете доступны следующие разделы: фамилия, имя, фотография, о себе, интересы, дата рождения, статус пользователя и в какой комнате он сейчас находится. Практически все данные сохраняются в базу и извлекаются из нее одним способом.

Из всех перечисленных данных, которые отображаются в анкете, из базы данных не извлекается только информация о фотографии. Так как чат не представляет собой социальную сеть, то было принято решение о хранении только одной фотографии для каждого пользователя, что значительно упрощает метод хранения фотографий на сервере и доступа к ним. Все они хранятся в папке info\_users/photo и имеют специальные названия: id.jpg, где ID – уникальный идентификатор пользователя. Таким образом, для того чтобы загрузить нужную нам фотографию необходимо знать только ID нужного пользователя.

**Система участников**

Для чата была составлена следующая система должностей (рисунок 2). На схеме видно, что пользователи с высоким приоритетом, имеют преимущества над пользователями, у которых приоритет ниже. Обратное невозможно. За каждой должностью закреплены свои права и обязанности.

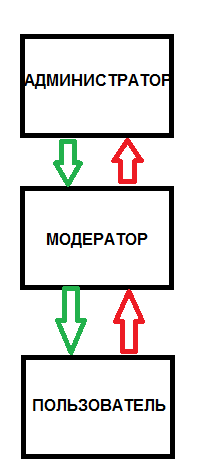


Рисунок 2 – Схема взаимодействия администрации и пользователей в чате

Администратор. Обладает возможностью доступа ко всем функциям чата. Должность администратора можно получить только, имея доступ к базе данных, сделано это из соображений безопасности.

Модератор. Имеет доступ только к некоторым функциям: удаление комнат, блокировка/разблокировка пользователей, принятие решения о смене имени и фамилии, блокировка фотографий пользователя, отправка пользователю системных сообщений.

Пользователь. Имеет доступ к созданию комнат (если его рейтинг позволяет это сделать), редактированию своей анкеты, общению в чате, изменению своих настроек.

**Рейтинг пользователя.**

Каждому пользователю доступна система рейтинга, благодаря которой пользователю открываются дополнительные функции в чате. Рейтинг пользователя зависит от его активности в чате. За каждые 10 сообщений рейтинг пользователя увеличивается на одну единицу. Таким образом, с помощью рейтинга можно отследить активность пользователя в чате.

**Панель управления.**

К данному разделу имеют доступ только те пользователи, приоритет которых больше нуля, т.е. модераторы и администраторы. В данном окне видны все функции, которые доступны модератору или администратору. Для каждой функции имеется подробное описание и случаи, когда можно использовать эту функцию (например, в каком случае возможна блокировка пользователя).

**3. РЕЗУЛЬТАТ РАБОТЫ**

В ходе разработки чата было получено готовое веб-приложение. Для того, чтобы убедиться, что скорость получившегося чата находится на высоком уровне и он полностью оптимизирован, воспользуемся специальным сервисом PageSpeed Insights от Google. Первым делом проверим, что наш чат сможет получить максимальную оценку:

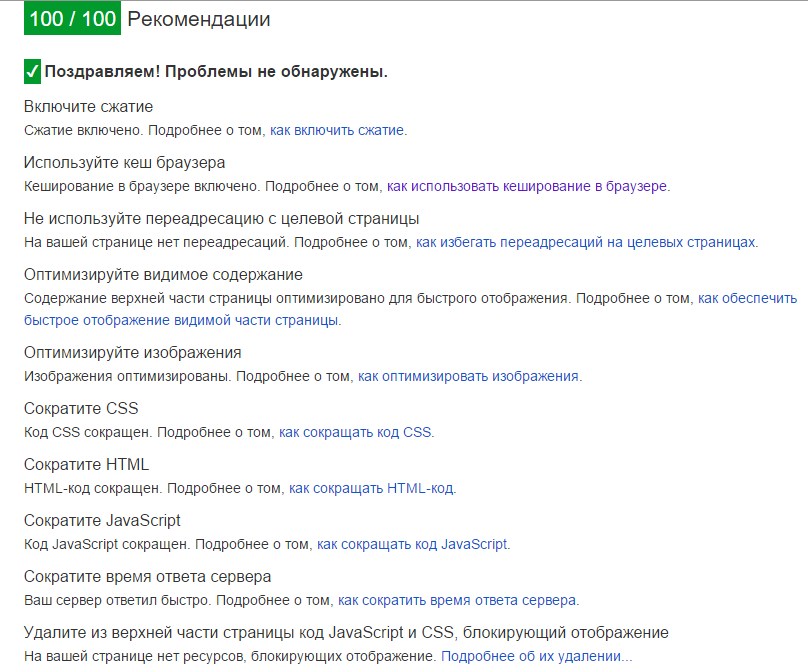


Рисунок 3 – Рейтинг разработанного чата на сервисе PageSpeed Insights

Теперь проверим рейтинги других чатов:

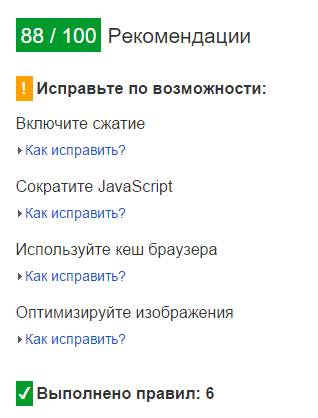


Рисунок 4 – Рейтинг чата tiamochat.ru на сервисе PageSpeed Insights

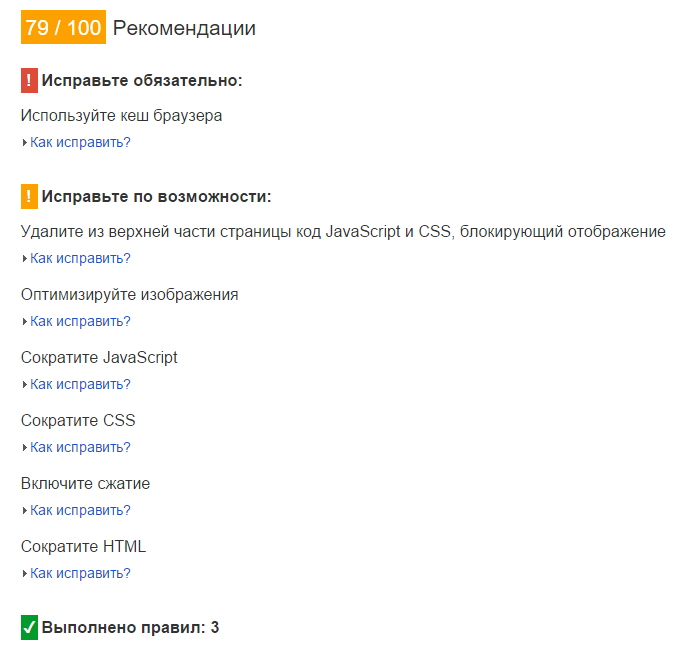


Рисунок 5 - Рейтинг чата 4brown.su на сервисе PageSpeed Insights

Как видно из результатов теста, то в рассмотренных чатах встречаются ошибки, которые влияют на скорость загрузки страниц. Таким образом, удалось создать веб-приложение с высокой скоростью загрузки страниц и большим функционалом.

+ ссылка на готовый чат

**4. ЗАКЛЮЧЕНИЕ**

В процессе разработки были приобретены навыки работы с скриптовым языком общего назначения – PHP, который широко применяется для разработки веб-приложений, поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков, применяющихся для создания динамических веб-сайтов. Были получены базовые знания языка JavaScript, который является прототипно-ориентированным сценарным языком программирования. Также была изучена библиотека JQuery языка JavaScript, которая фокусируется на взаимодействии JavaScript и HTML. В ходе разработки приложения, был получен опыт работы с базами данных, а также СУБД MySQL. Благодаря веб-приложению PHPMySQL были получены навыки администрирования СУБД MySQL. А благодаря технологии Ajax, были получены навыки обмена данными браузера и сервера в “фоновом” режиме. Таким образом, при создании чата, был получен большой опыт в создании сайтов и взаимодействии различных технологий и языков программирования.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Автор. Название статьи. В свободном доступе: <http://volchat.ru/>
2. <http://4brown.su/>
3. <http://tiamochat.ru/>
4. <http://ulogin.ru/>
5. <https://developers.google.com/>