МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение высшего образования   
**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**  
**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра вычислительных технологий**

**КУРСОВАЯ РАБОТА**

**ТЕХНОЛОГИИ РАЗРАБОТКИ WEB-ПРИЛОЖЕНИЙ**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М.В. Краснопер  
 (подпись)

Направление 02.03.02 «Фундаментальная информатика и информационные технологии»

Направленность (профиль) «Вычислительные технологии»

Научный руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Т.А. Приходько  
канд. техн. наук, доц. (подпись)

Нормоконтролер \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_А.А. Климец  
ассистент каф. ВТ (подпись)

Краснодар

2019

СОДЕРЖАНИЕ

[Введение 3](#_Toc9859669)

[1 Технологии разработки Web-приложений 4](#_Toc9859670)

[1.1 Клиентские Web-технологии 4](#_Toc9859671)

[1.2 Серверные Web-технологии 7](#_Toc9859672)

[1.2.1 Паттерны MVC, MVP и MVVM 7](#_Toc9859673)

[1.2.2 PHP vs Node.js vs Python 10](#_Toc9859674)

[1.2.3 MySQL vs PostgreSQL 12](#_Toc9859675)

[2 Практическая реализация Web-приложения 15](#_Toc9859676)

[2.1 Постановка задачи 15](#_Toc9859677)

[2.2 Реализация поставленной задачи 15](#_Toc9859678)

[Заключение 20](#_Toc9859679)

[Список использованных источников 21](#_Toc9859680)

[Приложение А index.php поиск по группе 22](#_Toc9859681)

[Приложение Б](#_Toc9859683) [index.php вызов модального окна 23](#_Toc9859684)

[Приложение В](#_Toc9859685) [index.php модальное окно добавления пары 25](#_Toc9859686)

[Приложение Г](#_Toc9859687) [handler.php модальное окно добавления пары 27](#_Toc9859688)

# ВВЕДЕНИЕ

Влияние World Wide Web на жизнь человека неизмеримо. Глобальная сеть Интернет прочно вошла в нашу жизнь, и нам уже трудно представить свою деятельность без нее. Web-технология полностью перевернула представление о работе с информацией и компьютером в целом. Оказалось, что важным критерием внедрения и развития новых технологий, помимо производительности, был механизм взаимодействия между человеком и компьютером. Как только интерфейс взаимодействия был упрощен до естественности восприятия обычным человеком, последовал взрыв интереса к возможностям вычислительной техники.

На сегодняшний день WWW предоставляет огромный набор возможностей, существуют миллионы сайтов, на которых размещена всевозможная и, что немаловажно, легкодоступная информация, а также сайты по продаже товаров и услуг. Любой человек, имея доступ в Интернет, может найти практически любую информацию и воспользоваться ей так, как ему захочется, всего за пару кликов. Могло ли человечество всего несколько десятков лет назад представить, что им не обязательно будет выходить на улицу, чтобы купить тот или иной товар или прочитать ту или иную книгу. Исходя из вышесказанного, не трудно понять, что создание Web-приложений является одним из важнейших вопросов разработки ресурсов глобальной сети Интернет.

# 1 Технологии разработки Web-приложений

Вся информация в Web-браузере отображается в виде Web-приложений (сайтов), которые являются основным элементом байтов памяти WWW. Web-приложения, поддерживая технологию мультимедиа, объединяют в себе различные виды информации: текст, графику, звук, анимацию и видео. От того, насколько качественно и красиво сделан тот или иной сайт, зависит во многом его успех в Сети. Пользователю приятно посещать те страницы, которые имеют стильное оформление, не наполнены чрезмерно графикой и анимацией, быстро загружаются и правильно отображаются в окне Web-браузера.

В настоящее время во всемирной паутине размещено несколько миллионов Web-сайтов и их число постоянно растет. Существует множество технологий и средств для создания Web-приложений, но лишь некоторые из них способны предоставить разработчикам инструменты для решения подавляющего большинства стоящих перед ними задач.

* 1. **Клиентские Web-технологии**

Рассмотрим следующие основные клиентские Web-технологии[1].

- HTML (HyperText Markup Language) – язык разметки, предназначенный для создания Web-приложений. Был изобретен и стандартизирован корпорацией W3C в 1996 году. HTML описывает структуру документа, определяя, например, блоки, текст, изображения, формы ввода данных, таблицы, но он не задает конкретных и точных атрибутов форматирования документа, этим занимается браузер. Не является языком программирования, но Web-приложения могут включать в себя встроенные программы-скрипты на других языках программирования, например на JavaScript и PHP.

- CSS (Cascading Style Sheets) – язык таблиц стилей, определяющий оформление и макет элементов HTML. Описывает правила форматирования элементов с помощью свойств и допустимых значений этих свойств. Для каждого элемента можно использовать ограниченный набор свойств. Если HTML можно назвать скелетом, то CSS – одеждой Web-приложения. Стили могут применяться непосредственно к элементу HTML, отдельно определяться на той же странице в контейнере <head>, используя специальный тег <style>, либо определяться в отдельном файле, ссылка на который также будет задаваться в контейнере <head>, используя <link>.

- JavaScript – динамически интерпретируемый язык программирования. При работе с ним в Web-приложении обычно требуется выполнять такие задачи, как:

а) обращение к любому HTML-элементу и проведение манипуляций с ним, таких как удаление, обновление, добавление и другие;

б) запуск код в соответствии с определенными событиями, например, нажатием на кнопку или любой другой элемент html;

в) запрос данных или отправка команды в Web-API.

Вышеперечисленные задачи также можно выполнять при помощи готовых библиотек, позволяющих значительно упростить это. Одной из самых первых и популярных таких библиотек является jQuery, которая по-прежнему используется разработчиками при работе с Web-приложениями.

Возможности JavaScript значительно расширяются при использовании так называемых API.

API – готовые наборы блоков кода, которые позволяют реализовывать программы, которые в противном случае было бы трудно или невозможно реализовать[2]. Обычно API делятся на 2 группы:

1. API браузера встроены в браузер и способны осуществлять сложные действия с различного вида данными, например:

а) API-интерфейс DOM позволяет манипулировать HTML и CSS, то есть создавать, удалять, изменять элементы HTML, динамически изменять стили каких-либо элементов, создавать всплывающие окна;

б) API геолокации извлекает географическую информацию;

в) API Canvas и WebGL позволяют создавать анимированные 2D и 3D-графики;

г) Аудио и видео API, такие как HTMLMediaElement и WebRTC, позволяют делать действительно интересные вещи с мультимедиа, такие как проигрывание аудио и видео прямо на веб-странице.

2) Сторонние API – устанавливаются отдельно и обладают собственным уникальным функционалом. Например, Google Maps API позволяет встраивать пользовательские карты на Web-сайте.

На этом возможности JavaScript не заканчиваются. На сегодняшний день данный язык очень популярен и стремительно развивается. Помимо написания клиентского кода, JavaScript используется также при написании серверного кода, о чем будет говориться дальше.

- Ajax (Asynchronous JavaScript and XML) – технология обращения к серверу без перезагрузки страницы, благодаря чему Web-приложение становится быстрее и удобнее. Другими словами, при использовании технологии Ajax скрипт вносит изменения только в ту часть документа, из которой пришел до этого запрос. Можно выделить 3 основных возможности данной технологии:

1) полезен при заполнении форм, например, при авторизации или нажатии на кнопку подписки, добавления в корзину и прочее,

2) динамическая подгрузка данных,

3) живой поиск, например, при вбивании слова в поиск в Google.

Удобно использовать в паре с вышеназванной библиотекой jQuery.

- Angular.JS – относительно новый JavaScript фреймворк с открытым исходным кодом, впервые представленный в 2009 году. Предназначен для разработки одностраничных приложений. Его цель – расширение браузерных приложений на основе MVC-шаблона, а также упрощение тестирования и разработки.

Одной из ключевых возможностей Angular.JS являются директивы, позволяющие разработчику описать поведение отдельных элементов и расширить синтаксис HTML. Помимо этого, Angular позволяет декларативно описывать как должен меняться UI в зависимости от изменения состояния приложения, что дает возможность не заниматься утомительной программной манипуляцией DOM, удалять шаблонный код (например, колбеки) без каких-либо потерь и сохранять только такой код, который описывает общий поток данных в Web-приложении.

Исходя из возможностей Angular.JS можно сказать, что данный фреймворк позволяет сделать из рутинных задач тривиальные, а из сложных – возможные. Кроме этого, порог входа в Angular.JS достаточно низок, а значит, что научиться работать с ним достаточно просто.

Помимо приведенных выше клиентских Web-технологий стоит также отметить такие популярные технологии, как React, Vue, Laravel и другие, предоставляющие нескромные возможности по разработке Web-приложений.

* 1. **Серверные Web-технологии**
     1. **Паттерны MVC, MVP и MVVM**

С течением времени разрабатываемые Web-приложения становятся больше и сложнее. Вместо одной сплоченной группы разработчиков, работающей над проектом, собирается много разных групп, каждая из которых отвечает за определенную область, будь то GUI, бизнес-логика или другие компоненты, стали использоваться специальные люди, ответственные за анализ и тестирование разрабатываемого Web-сайта или иного программного обеспечения, вследствие чего стоимость реализации таких задач существенно возрастает. Учитывая желание уменьшения трудозатрат на решение такого рода задач, становится ясно, что необходимо использовать готовые унифицированные решения – паттерны [3]. Рассмотрим три основных паттерна:

1. Model-View-Controller разделяет приложение на три основных части (см. рисунок 1): модель (model), представление (view) и контроллер (controller) – таким образом, что изменение любого компонента может осуществляться независимо друг от друга. Используется в ситуации, когда связь между представлением и другими частями невозможна.

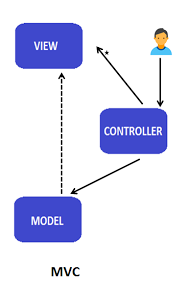


Рисунок 1 – структура MVC шаблона

Модель предоставляет данные и методы работы с ними, не зависит от представления и контроллера, то есть не знает, как визуализировать данные и не взаимодействует с пользователем. Представление отвечает за отображение пользователю необходимых данных на экране, может влиять на контроллер. Контроллер обрабатывает действия пользователя и определяет какое именно представление View должно быть отображено в данный момент.

1. Model-View-Presenter является производной от MVC и используется в основном для построения пользовательского интерфейса. Структура такого шаблона представлена на рисунке 2. MVP используется в том случае, когда у разработчика нет возможности связать данные [4].

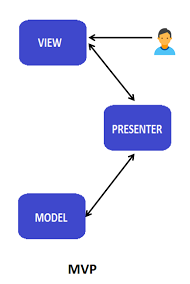


Рисунок 2 – структура MVP шаблона

Вместо контроллера здесь используется представитель (presenter), являющийся посредником между представлением и моделью. Его действия заключаются в получении входных данных от пользователя через представление (view), их обработки с помощью методов модели и передачи обратно пользователю в удобном для него виде через представление.

1. Model-View-View-Model поддерживает двустороннюю привязку данных между представлением (view) и моделью представления (View-model), что позволяет с изменением View-model автоматически изменять также и View (см. рисунок 3). MVVM используется в ситуации, когда у разработчика есть возможность связывания данных.

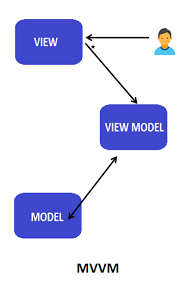


Рисунок 3 – структура MVVM шаблона

View-model не может обращаться напрямую к view. Вместо этого она представляет легко связываемые свойства и методы в виде команд. View может привязываться к этим свойствам и командам, чтобы получать информацию из View-model и вызывать на ней методы [5].

Подводя итоги, стоит отметить, что раньше для создания качественного и крупного Web-приложения могло хватить знаний HTML, CSS, JavaScript и PHP, однако сейчас на основе выше перечисленных шаблонов появляется возможность создавать более мощные приложения за то же время, если даже не быстрее. Такие технологии диктуют будущее Web-программирования, упрощая однотипную рутинную работу, и без них не обойтись.

* + 1. **PHP vs Node.js vs Python**

Существуют различные платформы для написания серверного кода, который будет задавать функционал нашего Web-приложения. В наше время сложно точно сказать, что именно подойдет лучшего всего для бэкэнда, так как у каждой платформы свои особенности, поэтому приведем анализ трех следующих платформ:

1. PHP – скриптовый язык, в основном используемый при создании Web-приложений

Приведем преимущества PHP [6].

- Обладает огромной базой для различных решений от систем управления контентом до мощных фреймворков, таких как Laravel или Symfony

- Достаточно независим от платформы и может быть запущен практически на любом сервере и любой платформе.

- PHP изначально был разработан для разработки Web-приложений, в отличие от JavaScript и Python, из-за чего обладает всеми необходимыми функциями для работы с HTML, серверами и базами данных.

Ниже приведены недостатки PHP.

- Не очень хорош в реализации паттерна MVC, который является рекомендованным в Web-разработке.

- PHP следует классической клиент-серверной модели где запрос страницы инициирует приложение, соединение с базой данных, их обработкой и рендерингом HTML, что делает PHP медленнее Node.js.

2) Node.js является средой выполнения JavaScript, позволяющая создавать масштабируемые приложения.

Приведем основные преимущества Node.js.

- Несмотря на однопоточность Node, его стандартная библиотека предоставляет модули масштабирования, с помощью которых появляется возможность создания нескольких Node процессов.

- Не имеет строгих правил и зависимостей, что дает волю креативности разработчиков.

- Один язык на бэкэнде и фронтэнде.

Ниже приведены недостатки Node.js.

- Малая эффективность в операциях, интенсивно использующих CPU, что обусловлено особенностями событийно-ориентированной архитектуры.

- Недостаточная или отсутствующая документация для некоторых модулей в npm.

3) Python - высокоуровневый язык программирования с акцентом на читабельность и продуктивность кода.

Приведем преимущества языка Python в Web-разработке [7].

- Благодаря краткому простому синтаксису и динамической типизации позволяет писать легко поддерживаемые и расширяемые программы.

- Быстрота разработки.

- Большая стандартная библиотека для бэкэнд разработки.

И, наконец, перейдем к недостаткам Python.

- Область применения языка ограничена серверными приложениями, его поддержка отсутствует на мобильных платформах и браузерах.

- Отсутствия поддержки DSL (Domain-specific Languages), который оптимизирует разработку UI.

Таким образом, из выше приведенного анализа сложно определить явного лидера, так как каждая платформа хороша для определенных проектов. Если требуется быстро развернуть производительное приложение, способное выдерживать большие нагрузки на CPU, то стоит выбрать Python. Если стоит цель написания приложения реального времени или одностраничного сайта, то концепция Node.js отлично подойдет для реализации такого вида задач. В случае, когда необходимо стандартизированное решение с обширной поддержкой хостинга, стоит уделить внимание PHP.

* + 1. **MySQL vs PostgreSQL**

Не менее важным этапом создания Web-приложения является выбор подходящей базы данных. В наше время существует много различных СУБД, в связи с чем возникает проблема выбора наиболее подходящей. Приведем анализ двух популярных СУБД - MySQL и PostgreSQL.

1. MySQL является общей системой управления реляционными базами данных с открытым исходным кодом (RDBMS) [8]. Такая база данных использует архитектуру клиент – сервер, состоящую из многопоточного сервера SQL. К основным характеристикам MySQL можно отнести следующее.

- Совместима с несколькими платформами.

- Обеспечивает поддержку управления MVCC (MultiVersion Concurrency Control).

- Соответствует стандартам ANSI SQL.

- Может обрабатывать данные в 50 миллионов и более строк.

- Объектно-ориентированная СУБД.

1. PostgreSQL является системой управления объектно-реляционными базами данных (ORDBMS). Помимо этого Postgre является транзакционной, ACID-совместимой, имеет материализованные и обновляемые представления, внешние ключи и триггеры. К основным характеристикам PostgreSQL можно отнести следующее.

- На данный момент активно развивается.

- Работает на основных платформах ОС.

- Поддержка JSON и XML.

- Обширная индексация.

- Наличие триггерных процедур.

Сравнивая MySQL и PostgreSQL можно выделить три ключевых пункта[9]:

1. PostgreSQL наиболее совместим с SQL, поскольку он соответствует 160 из 170 ключевых компонентов SQL-стандарта, а также нескольким нестандартным компонентам, в то время как MySQL только частично совместим с SQL и не применяет полный стандарт SQL.
2. PostgreSQL имеет встроенную поддержку SSL, роли и унаследованные роли для поддержки и создания разрешений доступа к базе данных, а также присутствует безопасность на уровне строк. MySQL же обеспечивает безопасность на основе списков контроля доступа ACL для всех запросов и соединений, но существует также поддержка SSL-зашифрованных соединений.
3. PostgreSQL рекомендуется использовать в больших системах, где скорость записи и чтения являются ключевыми параметрами, в то время как MySQL подходит для более простых проектов. При больших нагрузках или сложных запросах MySQL может работать медленнее, чем PostgreSQL.

Исходя из анализа обеих СУБД, можно сделать вывод, что PostgreSQL рекомендуется использовать при разработке большого проекта со сложной логикой и сложными связями между таблицами. Такая СУБД обеспечит более высокую производительность и гибкость разрабатываемого Web-приложения, нежели MySQL. Однако, если в целях не стоит создание всемирно известного огромного Web-приложения, MySQL может стать неплохим выбором, не обеспечивая разработчиков излишними характеристиками.

# 2 Практическая реализация Web-приложения

## 2.1 Постановка задачи

Цель практического исследования заключается в разработке базы данных и оболочки для отображения расписания факультета ФКТиПМ, формировании и выводе расписания для указанной группы, а также возможности редактирования данных, составляющими это расписание. Входные данные – расписание, сформированное специалистом по составлению расписаний в xls-формате. Для выполнения поставленной цели необходимо решить следующие задачи: для каждого дня создать таблицу, которая будет заполняться значениями из базы данных и добавить кнопки просмотра недели-числителя и недели-знаменателя, обеспечить возможность обработки события on-click (при нажатии на ячейки таблиц выполняется добавление, удаление или редактирования содержимого ячейки - описания пары).

В качестве языка программирования для написания клиентской части кода выступает JavaScript и его библиотека jQuery, для написания серверной части – PHP версии 7.4, база данных – MySQL.

## 2.2 Реализация поставленной задачи

Программу можно разделить на четыре части:

1. makson.db,
2. handler.php,
3. index.php,
4. styles.css.

Рассмотрим основные моменты каждой части программы подробнее.

1. Обязательной частью выполнения поставленной задачи является этап составления базы данных. Для этого воспользуемся Web-приложением phpMyAdmin, представляющим собой интерфейс для администрирования СУБД MySQL. Для начала нужно определить, какие данные статические, то есть можно задать их в программе в виде массивов, а какие могут меняться со временем. К статическим данным относятся такие классы информации, как день недели (day), четность пары (parity\_week), тип (lesson\_type) и номер пары (number). Остальные данные, такие как группа, предмет, преподаватель, подгруппа и кабинет, являются динамическими, так как может возникнуть потребность, например, добавить нового преподавателя или предмет, удалить пару или поменять кабинет. Создадим пять таблиц для наших динамических данных, в каждой таблице будет по два значения: id типа int и name типа varchar.

Дальше нужно составить таблицу, которая будет задавать все пары, и объединять в себе все выше перечисленные данные. Назовем эту таблицу lesson и зададим от таблиц groups, subjects, teachers, sub\_groups и cabinets связи 1 к 1, так как пару на текущий момент может вести только 1 преподаватель, только определенный предмет, в указанном кабинете для одной группы и, возможно, какой-то подгруппы. Структура получившейся базы данных представлена на рисунке 4.

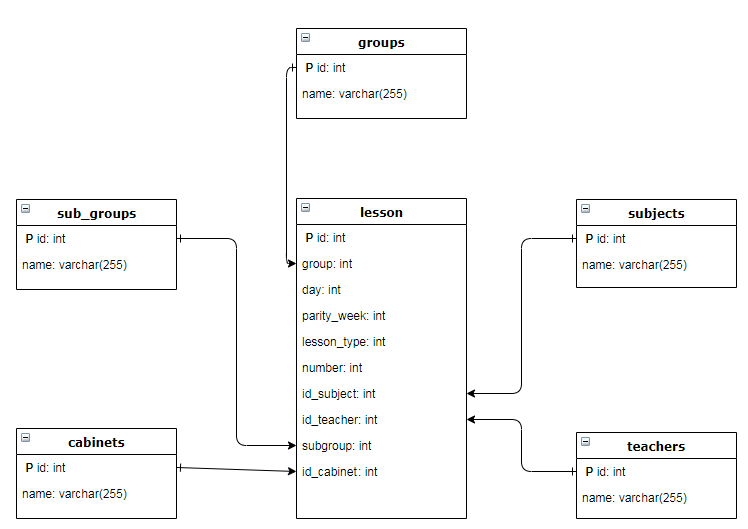


Рисунок 4 – структура базы данных расписания makson.db

1. handler.php содержит в себе серверную часть кода, которая описывает весь функционал нашего Web-приложения, такой как поиск пар по группе, добавление, удаление, изменение пары, отправка данных в модальное окно добавления или редактирования пары, вывод числителя или знаменателя на экран. Файл получает информацию с клиентской стороны приложения, выполняет действия, требующиеся от него, и отправляет результат обратно. На примере поиска пар по группам покажем, как устанавливается связь с базой данных с последующим выполнением необходимых запросов и формируется результат.

С клиентской стороны приходит информация, которая задает входные данные и определяет выполняемые действия. Перед отправкой какого-либо запроса в базу данных необходимо установить с ней соединение. Для этого создадим два дополнительных php файла: dbconfig.php и DB.php. Первый файл содержит в себе всю необходимую информацию для соединения, такую как имя хоста, имя базы данных, логин и пароль. Второй файл содержит класс DB с методом getDb, который осуществляет соединение по установленным в dbconfig.php данным средствами расширения PDO (PHP Data Objects), предоставляющим простой и универсальный интерфейс для доступа к различным базам данных. Теперь, соединение с базой данных будет иметь следующий вид:

$db = DB::getDb();

Далее выполняется запрос на получение всех пар по указанной группе и четности недели (по-умолчанию выводится числитель), отсортированных по дням недели, а затем по номеру пары. После создается строка $snuffeen, которая будет содержать в себе HTML-код таблиц расписания, заполненных по полученным данным, и отправляется в клиентскую часть. Стоит также отметить, что каждая ячейка каждой таблицы имеет собственный номер (id) и класс, по которым определяются заданные события и CSS-стили.

1. Index.php задает изначальную HTML-структуру Web-приложения, а также все события по нажатию на те или иные элементы в виде скриптов. Любые действия на странице основаны на обращении к определенным элементам по уникальным идентификаторам и динамическом обновлении при помощи технологии Ajax. Обращения по номерам или классам реализованы методами JavaScript-библиотеки jQuery. Библиотека позволяет работать с элементами DOM без особых сложностей, например, удалять, заменять на другие элементы, редактировать их содержимое или встроенные значения, такие как id или класс, изменять параметры CSS, задавать события на нажатие и многое другое.

Так как в задаче присутствует много функций, все мы их рассматривать не будем, но стоит уделить внимание поиску по группам и вызову модального окна добавления, удаления или редактирования пары.

При нажатии на кнопку поиска по группе вызывается функция groupSearch, формирующая набор данных data. Этот набор заполняется следующими данными: действие, которое нужно выполнить (в нашем случае - поиск по группе), номер группы, введенный в поле input и четность недели. Пример такого набора приведен ниже.

var data = {'type': 'groupSearch', 'group': $("#inputGroup").val(), 'parity': parity};

Поле ввода имеет собственный идентификатор ‘inputGroup’, метод val берет из него текст и записывает его в data[‘group’]. Далее с помощью технологий Ajax отправляется запрос в серверную часть приложения, где выполняются ранее описанные действия. Чтобы Ajax функционировал верно, необходимо указать параметры так, как указано ниже.

url: 'handler.php',

type: "POST",

dataType: "html",

data: data,

Поле url указывает файл, к которому будет произведен запрос. Если такого файла не существует или будет присутствовать ошибка на стороне сервера, то обращение не принесет результата. Type указывает тип запроса, он может быть GET (получить) и POST (отправить). Затем указывается тип возвращаемых данных и сам набор данных.

Если сервер принял запрос и успешно его обработал, то созданная нами строка $snuffeen в клиентской части помещается в переменную response, после чего вставляется в отведенный под таблицы расписания блок класса “.tables” так, как показано ниже.

$(".tables").empty();

$(".tables").append(response);

Работа с модальным окном реализована по похожему принципу. При нажатии на ячейку из нее берется вся информация и вставляется в окно. Если ячейка была пуста, то есть не было пары, то предоставляется возможность добавить новую пару, указав все необходимые параметры. Параметры хранятся либо в базе данных, либо в массивах самой программы, если это статические данные. Если же ячейка не была пустой, то есть мы нажали на пару, можно удалить или редактировать ее содержимое. При добавлении или редактировании пары все теги <p> заменяются на <select> и заполняются присущими им значениями из базы данных или массивов, после чего к ним применяется плагин Chosen и они приобретают удобный и красивый для использования вид. Новые поля выбора предоставляют возможность поиска подходящего значения посредством ввода текста. После того, как все параметры были указаны, и была нажата кнопка «Сохранить», при помощи Ajax также отправляется запрос серверу, откуда уже идет обращение к базе данных.

1. Styles.css содержит все стили, используемые в нашем Web-приложении. Здесь указаны значения множества параметров, таких как цвет фона и шрифта, размеры каждой ячейки таблицы и самой таблицы, заданы анимации кнопок, поля ввода и многое другое.

# ЗАКЛЮЧЕНИЕ

В ходе выполнения курсовой работы были рассмотрены и проанализированы различные технологии создания Web-приложений, принципы и особенности их применения, а также преимущества и недостатки.

В рамках проделанной работы были подробно изучены такие технологии, как HTML, CSS, JavaScript и библиотека jQuery, Ajax, MySQL и PHP, приведены их возможности и цели применения. Помимо этого было проведено сравнение трех платформ для работы в бэкэнде, таких как PHP, Node.js Python и трех различных баз данных – MySQL, PostgreSQL и MongoDB. В конечном результате было создано такое Web-приложение, которое является оболочкой расписания, предоставляя не только его визуальное представление, но и возможность динамического изменения, а именно, добавления, редактирования или удаления тех или иных пар.

Поскольку не все запланированные задачи, были реализованы в ходе курсовой работы, ниже приводится план перспективного развития созданного Web-приложения.

- Добавить возможность авторизации, а также шифрование и защиту данных.

- Реализовать адаптивную верстку.

- Внедрить возможность выводить расписание для заданного преподавателя и кафедры.

- Добавить возможность загружать расписание из excel-файла в базу данных.

- Конвертировать базу, находящуюся на сервере в формат, удобный для использования в мобильном приложении.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Кирилл Ширинкин «Самообразование веб-разработчика» – СПб.: БХВ-Петербург. - 2014 г.
2. Web tutorials [Электронный ресурс]. – URL: <https://www.tutorialspoint.com/web_development_tutorials.htm> (дата обращения 12.05.2019).
3. MVC, MVP and MVVM Design Pattern [Электронный ресурс]. – URL: <https://medium.com/@ankit.sinhal/mvc-mvp-and-mvvm-design-pattern-6e169567bbad> (дата обращения 14.05.2019).
4. Технологии разработки пользовательских интерфейсов [Электронный ресурс]. – URL: <https://www.osp.ru/os/2013/10/13039072/> (дата обращения 14.05.2019).
5. 14 Technologies Every Web Developer Should Be Able to Explain [Электронный ресурс]. – URL: <https://differential.com/insights/14-technologies-every-web-developer-should-be-able-to-explain/> (дата обращения 14.05.2019).
6. PHP vs Node.js [Электронный ресурс]. – URL: <https://ua-blog.com/node-js-vs-php-какую-платформу-выбрать-для-вашего/> (дата обращения 15.05.2019).
7. Python vs Node.js [Электронный ресурс]. – URL: <https://www.educba.com/python-vs-node-js/> (дата обращения 15.05.2019).
8. SQL против NoSQL на примере MySQL и MongoDB [Электронный ресурс]. – URL: <https://tproger.ru/translations/sql-vs-nosql/> (дата обращения 17.05.2019).
9. Сравнение MySQL и PostgreSQL [Электронный ресурс]. – URL: <https://losst.ru/sravnenie-mysql-i-postgresql> (дата обращения 17.05.2019).

**ПРИЛОЖЕНИЕ А**

**index.php поиск по группе**

function groupSearch(parity) {

var group = $("#inputGroup").val();

groupNum = group;

group = group.replace(/\s+/g, '');

if (group.length !== 0) {

var data = {'type': 'groupSearch', 'group': $("#inputGroup").val(), 'parity': parity};

$.ajax({

url: 'handler.php',

type: "POST",

dataType: "html",

data: data,

success: function (response) {

console.log("Success");

$(".tables").empty();

$(".tables").append(response);

$(".parity\_weeks").css('display', 'flex');

},

error: function () {

alert('Ошибка. Данные не переданы');

}

});

}

}

**ПРИЛОЖЕНИЕ Б**

**index.php вызов модального окна**

function modal(id, day, parity, subgroup, type\_lesson, parity\_info, number, groupID) {

$(".tableplace").on('click', '#' + id, function(event){

$('.overlay').css('pointer-events','none');

$('.modal\_close').css('pointer-events','none');

event.preventDefault();

$('.overlay').fadeIn(400,

function(){

$('.modal\_form').css('display', 'flex').animate({opacity: 1, top: '50%'}, 50);

$('.whatDay').replaceWith('<p class="whatDay" id="whatDay' + id + '">' + $('#table' + day + ' #tableDay').text() + '</p>');

$('.whatTeacher').replaceWith('<p class="whatTeacher" id="whatTeacher' + id + '">' + $('#teacher' + id).text() + '</p>');

if (subgroup === 1) {

$('.whatSubgroup').replaceWith('<p class="whatSubgroup" id="whatSubgroup' + id + '">' + subgroup + ' пг' + '</p>');

}else if (subgroup === 2){

$('.whatSubgroup').replaceWith('<p class="whatSubgroup" id="whatSubgroup' + id + '">' + subgroup + ' пг' + '</p>');

} else if (subgroup === 3){

$('.whatSubgroup').replaceWith('<p class="whatSubgroup" id="whatSubgroup' + id + '">' + 'Общая' + '</p>');

}

$('.whatCabinet').replaceWith('<p class="whatCabinet" id="whatCabinet' + id + '">' + $('#cabinet' + id).text() + '</p>');

$('.whatSubject').replaceWith('<p class="whatSubject" id="whatSubject' + id + '">' + $('#subject' + id).text() + '</p>');

$('.whatTime').replaceWith('<p class="whatTime" id="whatTime' + id + '">' + (number + 1) + ' пара (' + timesArray[number] + ')</p>');

if (type\_lesson === 1) {

$('.whatType').replaceWith('<p class="whatType" id="whatType' + id + '">' + 'Лекция' + '</p>');

}else if (type\_lesson === 2){

$('.whatType').replaceWith('<p class="whatType" id="whatType' + id + '">' + 'Практика' + '</p>');

}

if (parity\_info === 1) {

$('.whatParity').replaceWith('<p class="whatParity" id="whatParity' + id + '">' + 'Числитель' + '</p>');

} else if (parity\_info === 2){

$('.whatParity').replaceWith('<p class="whatParity" id="whatParity' + id + '">' + 'Знаменатель' + '</p>');

} else if (parity\_info === 3){

$('.whatParity').replaceWith('<p class="whatParity" id="whatParity' + id + '">' + 'Обе недели' + '</p>');

}

$('#updateLesson').replaceWith('<button id="updateLesson" onclick="updateLessonInDB(' + id + ',' + parity + ',' + groupID + ',' +

'' + day + ',' + subgroup + ',' + type\_lesson + ',' + number + ')">Изменить</button>');

$('#deleteLesson').replaceWith('<button id="deleteLesson" onclick="deleteLessonFromDB(' + id + ',' + parity +')" style="margin-left: 7.5%">Удалить</button>');

$('.overlay').css('pointer-events','auto');

$('.modal\_close').css('pointer-events','auto');});});

**ПРИЛОЖЕНИЕ В**

**index.php модальное окно добавления пары**

function addLessonInDB(id, day, parity, subgroup, number, groupID) {

var data = {'type':'demoAdd', 'id': id, 'day': day, 'subgroup': subgroup, 'number': number, 'parity-global': parity, 'groupID': groupID};

$.ajax({

url: 'handler.php',

type: "POST",

dataType: "html",

data: data,

success: function (response) {

console.log("Success");

$(".modal\_form").empty();

$(".modal\_form").append('<span class="modal\_close"><i class="fas fa-sign-out-alt"></i></span>');

$(".modal\_form").append(response);

$(".selectWhatTeacher").chosen({display\_disabled\_options: false});

$(".selectWhatSubgroup").chosen({display\_disabled\_options: false});

$(".selectWhatCabinet").chosen({display\_disabled\_options: false});

$(".selectWhatSubject").chosen({display\_disabled\_options: false});

$(".selectWhatType").chosen({display\_disabled\_options: false});

$(".selectWhatParity").chosen({display\_disabled\_options: false});

$('.modal\_close, .overlay').click( function(){

$('.modal\_form')

.animate({opacity: 0, top: '45%'}, 50,

function(){

$(this).css('display', 'none');

$('.overlay').fadeOut(400);

console.log("Exited");

$(".modal\_form").empty();

$(".modal\_form").append('<span class="modal\_close"><i class="fas fa-sign-out-alt"></i></span><p class="whatDay"></p>' +

'<p class="whatTeacher"></p> <p class="whatSubgroup"></p> <p class="whatCabinet"></p> <p class="whatSubject"></p> ' +

'<p class="whatTime"></p> <p class="whatType"></p> <p class="whatParity"></p> ' +

'<button id="addLesson" style="display: none; width: 50%;">Добавить</button> <button id="updateLesson">Изменить</button> ' +

'<button id="deleteLesson" style="margin-left: 7.5%">Удалить</button>');

$('.overlay').css('pointer-events','auto');

$('.modal\_close').css('pointer-events','auto');

}

);

});

},

error: function () {

alert('Ошибка. Данные не переданы');

}

});

}

**ПРИЛОЖЕНИЕ Г**

**handler.php модальное окно добавления пары**

elseif ($\_POST['type'] == 'demoAdd'){

$demo\_id = $\_POST['id'];

$day = $\_POST['day'];

$subgroup = $\_POST['subgroup'];

$number = $\_POST['number'];

$global\_parity = $\_POST['parity-global'];

$groupID = $\_POST['groupID'];

$selects = '';

$db = DB::getDB();

$result = $db->prepare('SELECT \* FROM groups ORDER BY id ASC');

$result->execute();

$result->setFetchMode(\PDO::FETCH\_ASSOC);

$groupsList = $result->fetchAll();

$result = null;

$result = $db->prepare('SELECT \* FROM subjects ORDER BY id ASC');

$result->execute();

$result->setFetchMode(\PDO::FETCH\_ASSOC);

$subjectsList = $result->fetchAll();

$result = null;

$result = $db->prepare('SELECT \* FROM teachers ORDER BY id ASC');

$result->execute();

$result->setFetchMode(\PDO::FETCH\_ASSOC);

$teachersList = $result->fetchAll();

$result = null;

$result = $db->prepare('SELECT \* FROM sub\_groups ORDER BY id ASC');

$result->execute();

$result->setFetchMode(\PDO::FETCH\_ASSOC);

$subsList = $result->fetchAll();

$result = null;

$result = $db->prepare('SELECT \* FROM cabinets ORDER BY id ASC');

$result->execute();

$result->setFetchMode(\PDO::FETCH\_ASSOC);

$cabinetsList = $result->fetchAll();

$result = null;

$daysList = array('Понедельник', 'Вторник', 'Среда', 'Четверг', 'Пятница', 'Суббота');

$parityList = array('Числитель', 'Знаменатель', 'Обе недели');

$typesList = array('Лекция', 'Практика');

$timesArray = array('8:00 - 9:30','9:40 - 11:10','11:30 - 13:00','13:10 - 14:40','15:00 - 16:30','16:40 - 18:10','18:20 - 19:50');

$selects .= '<p class="whatDay" id="whatDay' . $demo\_id . '">' . $daysList[$day - 1] . '</p>';

$selects .= '<div id="divForSelectTeachers" class="divTeachers">';

$selects .= '<select id="teachersList" class="selectWhatTeacher">';

$selects .= '<option selected disabled hidden value="0">Преподаватель</option>';

foreach ($teachersList as $keyt => $teacher){

$selects .= '<option value="' . $teacher["id"] . '">' . $teacher["name"] . '</option>';

}

$selects .= '</select>';

$selects .= '</div>';

if ((int)$subgroup === 3) {

$selects .= '<div id="divForSelectSubgroup" class="divSubgroups">';

$selects .= '<select id="subgroupsList" class="selectWhatSubgroup">';

$selects .= '<option selected disabled hidden value="0">П/гр</option>';

foreach ($subsList as $keysu => $sub) {

if ((int)$sub["id"] != 3)

$selects .= '<option value="' . $sub["id"] . '">' . $sub["name"] . '</option>';

else

$selects .= '<option value="' . $sub["id"] . '">' . $sub["name"] . '</option>';

}

$selects .= '</select>';

$selects .= '</div>';

} elseif ((int)$subgroup === 1){

$selects .= '<div id="divForSelectSubgroup" class="divHoldSubgroup">';

$selects .= '<p class="whatHoldSubgroup" id="whatSubgroup' . $demo\_id . '" style="padding-top: 1%">1 пг</p>';

$selects .= '</div>';

} elseif ((int)$subgroup === 2){

$selects .= '<div id="divForSelectSubgroup" class="divHoldSubgroup">';

$selects .= '<p class="whatHoldSubgroup" id="whatSubgroup' . $demo\_id . '" style="padding-top: 1%">2 пг</p>';

$selects .= '</div>';

}

$selects .= '<div id="divForSelectCabinets" class="divCabinets">';

$selects .= '<select id="cabinetsList" class="selectWhatCabinet">';

$selects .= '<option selected disabled hidden value="0">Ауд</option>';

foreach ($cabinetsList as $keyca => $cabinet) {

$selects .= '<option value="' . $cabinet["id"] . '">' . $cabinet["name"] . '</option>';

}

$selects .= '</select>';

$selects .= '</div>';

$selects .= '<div id="divForSelectSubjects" class="divSubjects">';

$selects .= '<select id="subejctsList" class="selectWhatSubject">';

$selects .= '<option selected disabled hidden value="0">Предмет</option>';

foreach ($subjectsList as $keys => $subject) {

$selects .= '<option value="' . $subject["id"] . '">' . $subject["name"] . '</option>';

}

$selects .= '</select>';

$selects .= '</div>';

$selects .= '<div id="divForSelectTime" class="divTime">';

$selects .= '<p class="whatTimeAdd" id="whatTime' . $demo\_id . '">' . ($number + 1)." пара (".$timesArray[$number].")" . '</p>';

$selects .= '</div>';

$selects .= '<div id="divForSelectTypes" class="divTypes">';

$selects .= '<select id="typesList" class="selectWhatType">';

$selects .= '<option selected disabled hidden value="0">Тип</option>';

foreach ($typesList as $keyt => $type) {

$selects .= '<option value="' . ($keyt + 1) . '">' . $type . '</option>';

}

$selects .= '</select>';

$selects .= '</div>';

$selects .= '<div id="divForSelectParity" class="divParity">';

$selects .= '<select id="parityList" class="selectWhatParity">';

$selects .= '<option selected disabled hidden value="0">Неделя</option>';

foreach ($parityList as $keyp => $parity) {

$selects .= '<option value="' . ($keyp + 1) . '">' . $parity . '</option>';

}

$selects .= '</select>';

$selects .= '</div>';

$selects .= '<button id="saveLesson" onclick="save(' . $day . ',' . $global\_parity . ',' . (int)$subgroup . ',' . $number . ',' . $groupID . ')">Сохранить</button>';

$selects .= '<button id="cancel" onclick="cancel()" style="margin-left: 7.5%">Отменить</button>';

echo $selects;

}