МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное

бюджетное образовательное учреж­­дение высшего образования

«Дагестанский государственный университет»

Факультет Информатики и Информационных Технологий

Кафедра прикладной информатики

**КУРСОВОЙ ПРОЕКТ ПО ДИСЦИПЛИНЕ:**

Проектный практикум

«Создание web-сайта научно-технического работника ВУЗА»

Выполнил:

студент 3 курса очной формы обучения

по направлению 09.09.03 ПИЭ

Дегтярь Сергей Игоревич

Научный руководитель:

Камилов М-К. Б.

Работа допущена к защите:

Научный руководитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022г.

Махачкала, 2022­

Оглавление

[АННОТАЦИЯ 3](#_Toc101260602)

[ВВЕДЕНИЕ 4](#_Toc101260603)

[Глава 1 «АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ» 6](#_Toc101260604)

[1.1 Описание организации 6](#_Toc101260605)

[1.1.1 Анализ деятельности 6](#_Toc101260606)

[1.1.2 Требования к ИС 7](#_Toc101260607)

[ПЛАНИРОВАНИЕ 9](#_Toc101260608)

[2.1 Планирование сроков 9](#_Toc101260609)

[2.2 Планирование бюджета 11](#_Toc101260610)

[ПЕРЕЧЕНЬ ГРАФИЧЕСКИХ МАТЕРИАЛОВ 13](#_Toc101260611)

[3.1 Эскизный проект 13](#_Toc101260612)

[3.2 Структурная схема 17](#_Toc101260613)

[Глава 2 «РАЗРАБОТКА WEB-САЙТА» 17](#_Toc101260614)

[4.1 Технология создания сайта 17](#_Toc101260615)

[4.1.1 PHP-скрипт 18](#_Toc101260616)

[4.1.2 JavaScript 20](#_Toc101260617)

[4.1.3 HTML 20](#_Toc101260618)

[4.1.4 СУБД и MySQL 21](#_Toc101260619)

[4.1.5 Bootstrap 23](#_Toc101260620)

[4.2 Начало разработки 23](#_Toc101260621)

[4.2.1 Корневая структура проекта 23](#_Toc101260622)

[4.2.2 Создание БД 24](#_Toc101260623)

[4.2.3 Создание страниц (PHP) 25](#_Toc101260624)

[4.2.4 Live\_Search (JavaScript) 27](#_Toc101260625)

[Руководство пользователя 29](#_Toc101260626)

[ЗАКЛЮЧЕНИЕ 32](#_Toc101260627)

[СПИСОК ИСТОЧНИКОВ 33](#_Toc101260628)

# 

# АННОТАЦИЯ

В данном курсовом проекте проведён предметный анализ работы web-системы для научного работника ВУЗа. Описаны требования к информационной системе, структурной схеме. Рассмотрены основные принципы использования новейших технологии в области web-программирования, а также создание и работа c СУБД при помощи phpMyAdmin. Проведено планирование проекта и бюджета. Разработка интуитивно понятного эскиза проекта и реализован на системе Bootstrap. Указаны и описаны все использующиеся технологии и приведены их примеры непосредственно из кода проекта.

Для написания кодовой части был использован редактор кода Microsoft Visual Studio Code.

Для разработки эскизной части был использован графический редактор Figma.

Сайт реализован и протестирован на локальном сервере Open Server. Загрузкана web-сервер и установка домена не предусмотрена.

# ВВЕДЕНИЕ

За последнее десятилетие сеть интернет претерпела существенные изменения. Теперь сайты, ранее являвшиеся платформой для размещения статического контента, стали многофункциональными, интерактивными системами по предоставлению различной информации.

Актуальность данной работы обусловлена необходимостью анализа существующих методов разработки корпоративных и информационных ресурсов для выявления наиболее эффективного способа их создания.  Целью данного исследования является определение эффективных методов создания для грамотного построения корпоративных и информационных интернет-ресурсов. Под методами создания интернет-сайтов подразумевается совокупность приемов и инструментов разработки.

Как правило, все методики по разработке веб сайтов делятся на две группы. В первую группу входят способы изготовления сайтов ручного написания на основе использования языка разметки HTML, а также различных языков программирования. Для написания статического сайт необходимо сочетание HTML и CSS. Однако если нужен более сложный, динамический сайт, то тут подключают языки программирования. Основополагающими языками программирования, которые наиболее часто используются в разработке сайтов, являются PHP, Java и ASP.NET. Вторая группа методов разработки сайта проще в исполнении. Это очень просто, так как она не требует от программиста глубоких знаний в области программирования. В распоряжении разработчика имеется возможность через графический интерфейс специальных программ и CMS-систем создавать сайты.

При создании сайтов не стоит забывать об эстетической стороне. Именно поэтому пользователь в первую очередь обращает внимание не на сложность сайта, а на его дизайн и удобство пользования. Дизайн – это то, что пользователь видит первым. В случае если пользователь не может найти нужную ему информацию из-за сложностей навигации, он переходит на другой сайт. Чтобы успешно взаимодействовать со своим сайтом, необходимо его разработать с учетом эргономики. Эргономика обеспечивает комфорт во время чтения страницы, что позволяет удержать пользователя.

Данная курсовая работа посвящена изучению методов создания веб сайта для научного работника в формате блоко-блоговской системы. А также курсовая работа посвящена реализации собственного веб-сайта, позволяющего автору своего профильного предмета, делиться своими заданиями и лекционными файлами с другими пользователями своего блога.

# **Глава 1 «АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ»**

**1.1 Описание организации**

В рамках курсовой работы мной было принято начать разрабатывать систему веб-сервиса, который позволит квалифицированному педагогу в быстром порядке завести собственную страничку.

Так как мы будем использовать принцип блог-платформы, стоит обозначить этот термин для чёткого понимая.

Блог — веб-сайт, основное содержимое которого — регулярно добавляемые пользователем записи, содержащие текст, изображения или мультимедиа.

Для определения области я буду оперировать термином «блог», так как он лучше всего подходит под описание области.

Педагогический блог - это многостраничный, структурированный web-сайт, разработанный с помощью системы управления содержимым сайта (англ. Content Management System, CMS). Задачей такого сайта является предоставление возможности пользователю ознакомиться с полной информацией о преподавателях, открытых материалах, достижениях, а также продемонстрировать карьерную прогрессию.

### **1.1.1 Анализ деятельности**

Разрабатываемый web-сайт структурирует объёмы информации, связанные с автором страницы, а именно: информацию о авторе, его личных ресурсах и также полную педагогическую информацию.

Для создания веб-сайта выбран путь создания cms с нуля, написанный на скриптовом языке РНР. В системе применяются CSS-стили технологии (bootstrap), позволяющие мгновенно менять внешний вид сайта, а также доступно множество готовых тем -- наборов стилей для оформления текста и страницы в целом. Имеются все необходимые графические элементы дизайна, наборы скриптов и специальных дополнений -- виджетов, помогающих создать на сайте меню, удобную систему рубрик, архивы записей, поиск по сайту и прочие дополнительные удобства.

Также разработана поддержка возможности создания пользователей и определения их прав доступа при редактировании сайта.

Администратор сайта имеет доступ ко всем разделам каталога и может самостоятельно осуществлять его редактирование, также имеет возможность работы с файлами системы управления содержимым сайта.

Автор страницы имеет доступ только к изменению содержимого личной страницы, добавлению материалов и редактированию комментариев.

### **1.1.2 Требования к ИС**

ТРЕБОВАНИЕ К СТРУКТУРЕ ПОРТАЛА

Первоначальная структура сайта должна иметь следующий вид:

* Страница приветствия (Главная)
* Страница педагогов (Педагоги)
* Страница библиотека
* Личный кабинет
* Помощь

Основное меню:

ГЛАВНАЯ

* Стильное оформление
* Вводное пояснение
* Блоки с регистрацией и «Создать страницу»

ПЕДАГОГИ

* Отображение всех существующих созданных страничек
* Быстрый поиск на технологии Live\_Search

БИБЛИОТЕКА

* Отображение всех существующих файлов на сервисе для быстрого поиска нужной информации
* Быстрый поиск на технологии Live\_Search

ЛИЧНЫЙ КАБИНЕТ

* Раздел доступен для зарегистрированных пользователей.
* В кабинете имеется учетная информация пользователя, управление профилем, редактирование информации.

РЕГИСТРАЦИОННАЯ ФОРМА ПОЛЬЗОВАТЕЛЯ (\*-обязательно для заполнения):

* Фамилия \* — текстовое поле
* Имя \* — текстовое поле
* Отчество \* — текстовое поле
* Пароль \* — текстовое поле
* Повторить пароль \* — текстовое поле
* E-mail\* — (выступает логином)

ПОМОЩЬ

* Технические вопросы

*Требования к дизайну сайта:*

При разработке сайта должны быть использованы преимущественно светлые стили.

Основные разделы сайта должны быть доступны с первой страницы.

На первой странице не должно быть большого объема текстовой информации.

В дизайне сайта не должны присутствовать:

- мелькающие баннеры;

- много сливающегося текста;

- перегруз информации вне категории контента.;

- тёмные и агрессивные цветовые сочетания и графические решения

*Функциональные требования:*

Администратор

* Просмотреть личные данные всех пользователей
* Редактировать личные данные всех пользователей

Пользователь

* Просмотреть собственные личные данные.
* Редактировать: собственные личные данные.

Пользователь может редактировать только собственные данные:

* Фамилия\*
* Имя\*
* Отчество\*
* Страна\*
* Фото
* Информация о пользователе
* Языки

Авторизация

Пользователи могут авторизоваться на любой странице портала с помощью

собственной формы авторизации.

Форма содержит:

* Текстовое поле для ввода логина и пароля пользователя
* Текстовое поле для ввода ФИО
* Кнопку отправки формы.

Данные для доступа (авторизации):

* Логин — адрес электронной почты пользователя или никнейм
* Пароль — строка, содержащая от 8 символов, состоящая из A-z, 0-9.

Ниже формы располагаются ссылка:

* Забыли пароль

Форма «Забыли пароль» содержит поля:

* Email адрес пользователя, указанный при регистрации

При неудачной попытке авторизации появляется приглашение для повторной попытки авторизоваться с формой авторизации.

# **ПЛАНИРОВАНИЕ**

## **2.1 Планирование сроков**

Чтобы правильно рассчитать сроки реализации данного проекта нужно понимать количество основных этапов реализации. Опишем основные этапы и рассчитаем приблизительные сроки. Всего их будет 7.

**1 этап — Определение целей разработки сайта, проведение исследований**

На данном этапе создания сайта очень важно понять, зачем и для кого мы разрабатываем web-ресурс. Без верно поставленных целей и задач мы не сможем создать необходимый сайт, но даже в процессе создания вы поймете, что получите не то, что хотели. В нашем случае мы создаём концепцию сервиса предоставляющий бесплатную возможность создания узко тематического сайта профильных педагогических направлений. Также не маловажно проведение исследований конкурентов вашей тематики и составление схемы: какой функционал они предоставляют пользователям, какой дизайн и структуру имеют.

**2 этап — Разработка технического задания (ТЗ)**

Как правило, ТЗ может составлять как заказчик, так и исполнитель. Но обсуждается и согласовывается техническое задание, безусловно, обеими сторонами, т.к. какие-то вещи знать не может заказчик, а какие-то исполнитель. Составление правильного ТЗ просто необходимый шаг в этапах создания сайта, если что-то упустить в задании, например, дополнительный модуль, то исполнитель может отказаться от доработки (в рамках данной задачи).

Что должно включать в себя ТЗ:

* Обозначение целевых клиентов сайта, а также общую миссию;
* Структуру в виде схемы, состоящей из основных разделов, подразделов и примерного количества страниц;
* Пожелания к модулям (их великое множество: обратная связь, мы вам перезвоним, вопрос-ответ, фильтры и тому подобное);
* Описание дизайна (общее оформление — можно на примерах других сайтов, основные цвета, логотип, местонахождение различных блоков);
* Какие технологии использовать (вид CMS, библиотеки скриптов, будет ли мобильная версия и проч.);
* Порядок предоставления, обработки или создания графической и текстовой информации;
* Технические требования к сайту.

**3 этап — Создание дизайн-макета сайта**

В процессе разработки сайта при создании дизайна необходимо данный пункт разделить на несколько подэтапов создания:

* Предоставление оговоренного количества различных по концепции макетов главной страницы.
* Доработка и согласование выбранного варианта концепции.
* Создание макетов внутренних страниц (категории, товара / услуги, контактов, если планировалась мобильная версия, то каждый макет должен быть переделан и под мобильную версию).
* Доработка и согласование каждой из типовых страниц.

На макете может отсутствовать наполнение: как текстовая информация, так и фотографии, которые должен предоставить заказчик в процессе работ по созданию сайта — его временно можно заполнить демо текстом и картинками. Но все элементы дизайна должны быть прорисованы.

**4 этап — Верстка**

На данном этапе происходит чисто техническая работа нашими верстальщиком и программистом:

* "Понятный" код — чем код будет меньше и грамотнее, тем сайт загрузится быстрее.
* Валидность (корректность) — позволит без труда разобраться с кодом стороннему специалисту.
* Кроссбраузерная верстка (Cross-browser) — адекватное отображение сайта во всевозможных браузерах (IE, Chrome, Firefox, Opera и т.д.).

**5 этап — Программирование и натяжка на CMS**

CMS — это система управления сайтом, включающая в себя административную панель, которая, в свою очередь, располагает функциями управления контентом на сайте. Контент через админ-панель загружается очень просто, как документ Word. Натяжка включает в себя настройку вёрстки под параметры движка, собственно, сам CSS и путь к модулям движка в HTML.

В результате нескольких этапов работ мы получим полностью рабочий сайт, но без какого-либо наполнения.

Зачастую сайты создаются на готовых двигателях и это значительно облегчают работу исполнителю, исполнения данного проекта коррелирует в рамках от 2-ух до 5-месяцев с учётом использования готовых скриптовых решений и опыта исполнителей. В рамках курсовой работы период реализации определяется строгими временными ограничениями в виде месяца.

## **2.2 Планирование бюджета**

Так как мы определились с основными этапами на стадии архитектуры проекта, с кол-вом ресурсов (время). В рамках курсовой работы сайт будет реализован на локальном сервере ЭВМ, а значит все затраты будут определяться лишь во времени разработки проекта.

Взять во внимание реализация такого проекта за пределами научной работы может достигать от 20 до 30 тысяч рублей с учётом реализации интеграционных систем google или яндекс, установка на сервер, настройка оборудований, оплата хостинга и подключение доменного имени включая.

Основным сервисным расходом можно считать продвижение сайта «SEO».

*Из чего состоит продвижение сайта?*

* Работа над оптимизацией ресурса состоит не только из действий SEO-специалиста. Многие заказчики ошибаются с бюджетированием, потому что включают в SEO не все процессы, связанные с продвижением, а значит и не все расходы.

*Из чего состоит SEO:*

* Менеджмент всего процесса
* Координирование специалистов, составление технических заданий, контроль процесса.
* Оптимизация
* Работа SEO-специалиста на сайте.
* Автоматизация
* Дополнительные инструменты для ускорения работы оптимизатора.
* Разработка
* Программирование, изменения на сайте, контроль работы ресурса.
* Контент
* Работа над контентом: текстами, изображениями, наполнением карточек товаров.
* Аналитика
* Оценка результатов, прогнозирование, постановка цели.

*Статьи расходов на SEO:*

* Работа оптимизатора.
* Внесение изменений на сайт разработчиком.
* Работа службы контента.
* Софт для автоматизации SEO.
* Конверсионная оптимизация как отдельный постоянный объем работы.

# **ПЕРЕЧЕНЬ ГРАФИЧЕСКИХ МАТЕРИАЛОВ**

## **3.1 Эскизный проект**

Так как мы определились с общей структурой проекта, количеством страниц и контентом, стоит заняться разработкой эскизного проекта как прототипа нашего будущего сайта. Создание эскизного проекта будет осуществляться при помощи программы Figma, также можно использовать более «народную» программу Photoshop.

В рамках ограниченного времени будет нарисовано и реализовано 6 основных страницы:

1. Регистрация

2. Авторизация

3. Главная страница

4. Педагоги

5. Страница педагога

6. Личный кабинет (личная страница)

**РЕГИСТРАЦИЯ**

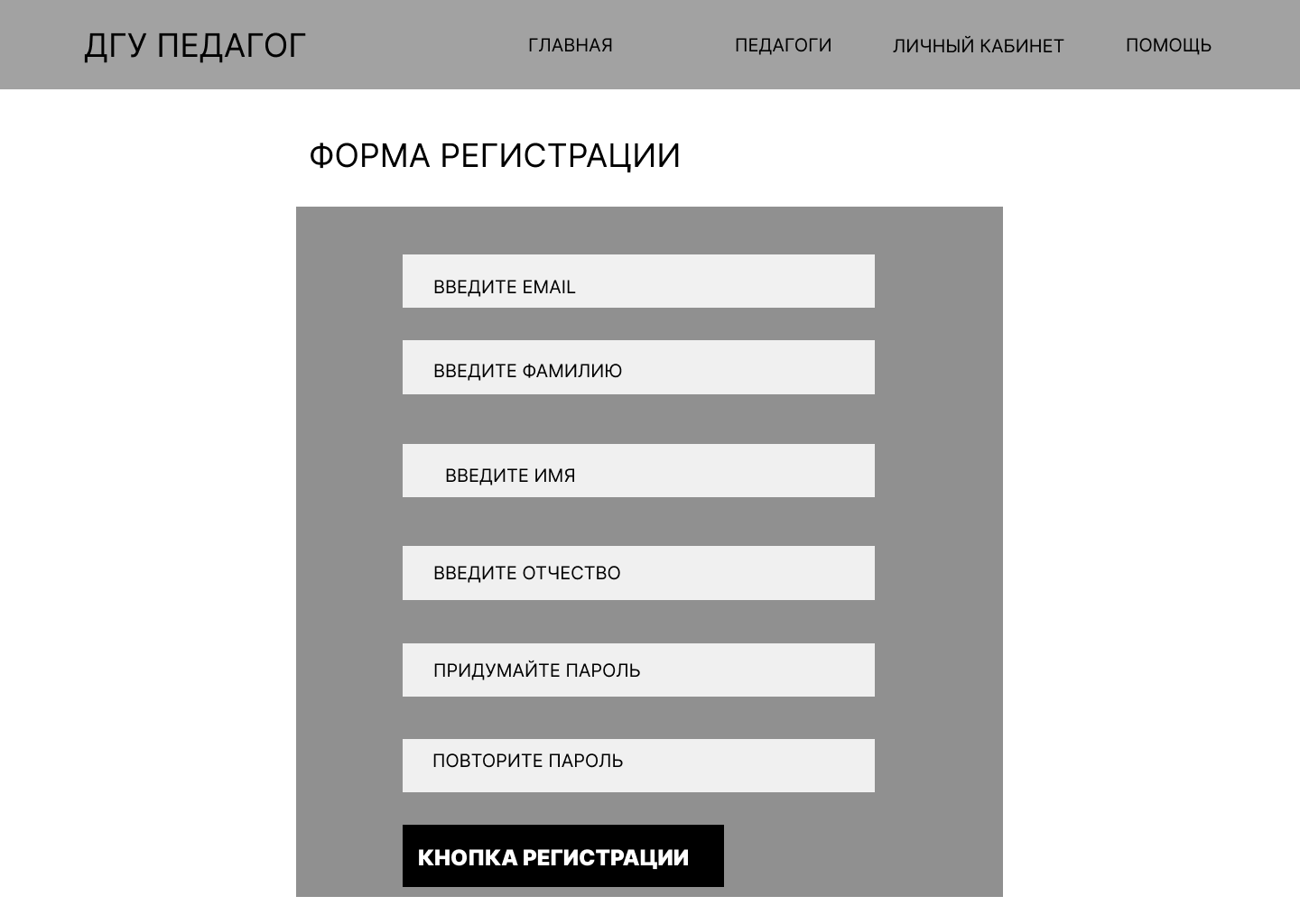


Рисунок 1 Регистрация

На рисунке 1 изображён процесс регистрации пользователей (педагогов) с дальнейшим занесением данных в БД. Все поля являются обязательным на этом этапе, но в дальнейшем они могут изменяться.

**АВТОРИЗАЦИЯ**

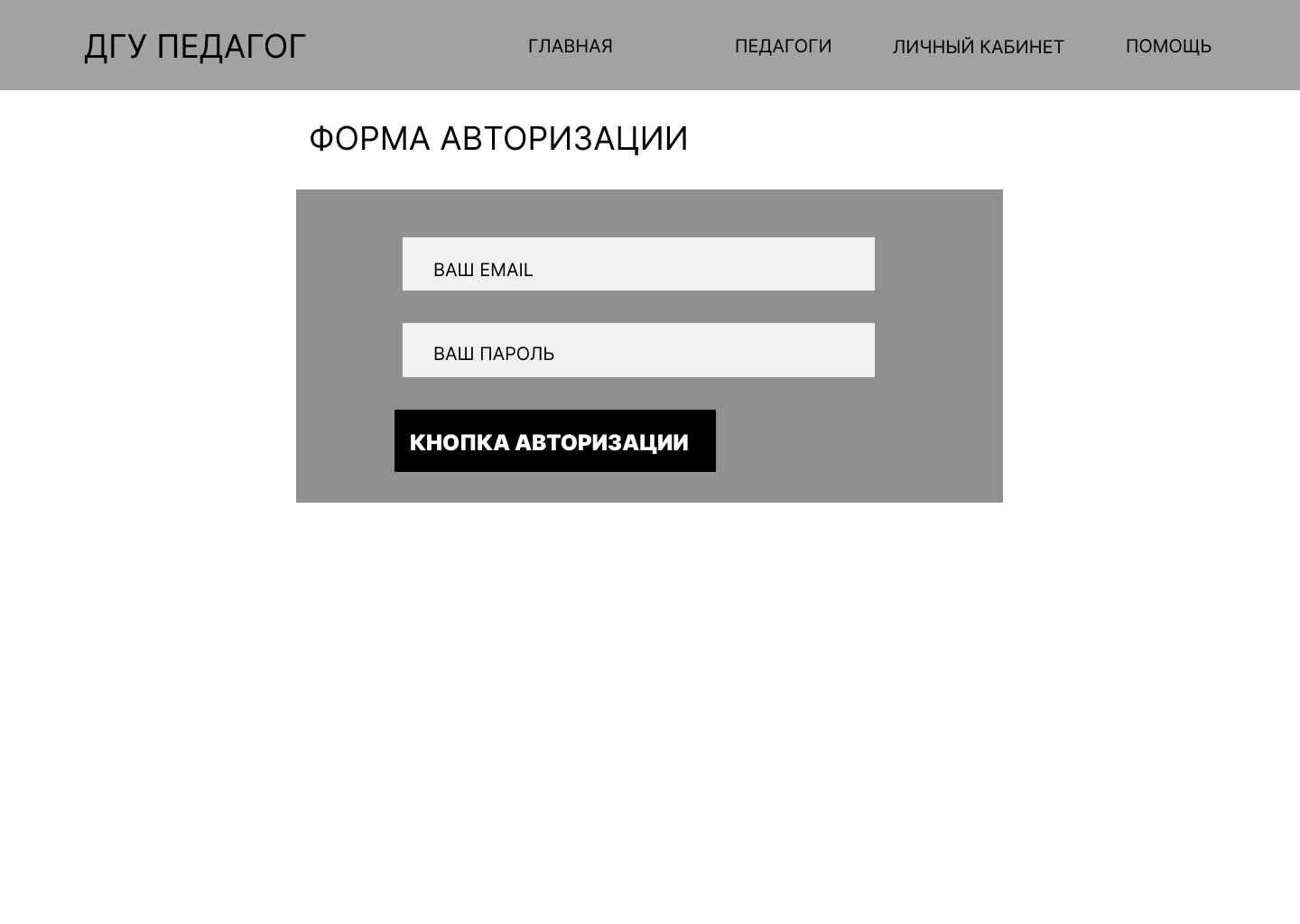


Рисунок 2 Авторизация

**ГЛАВНАЯ СТРАНИЦА**

Рисунок 3 Главная страница

**ПЕДАГОГИ**

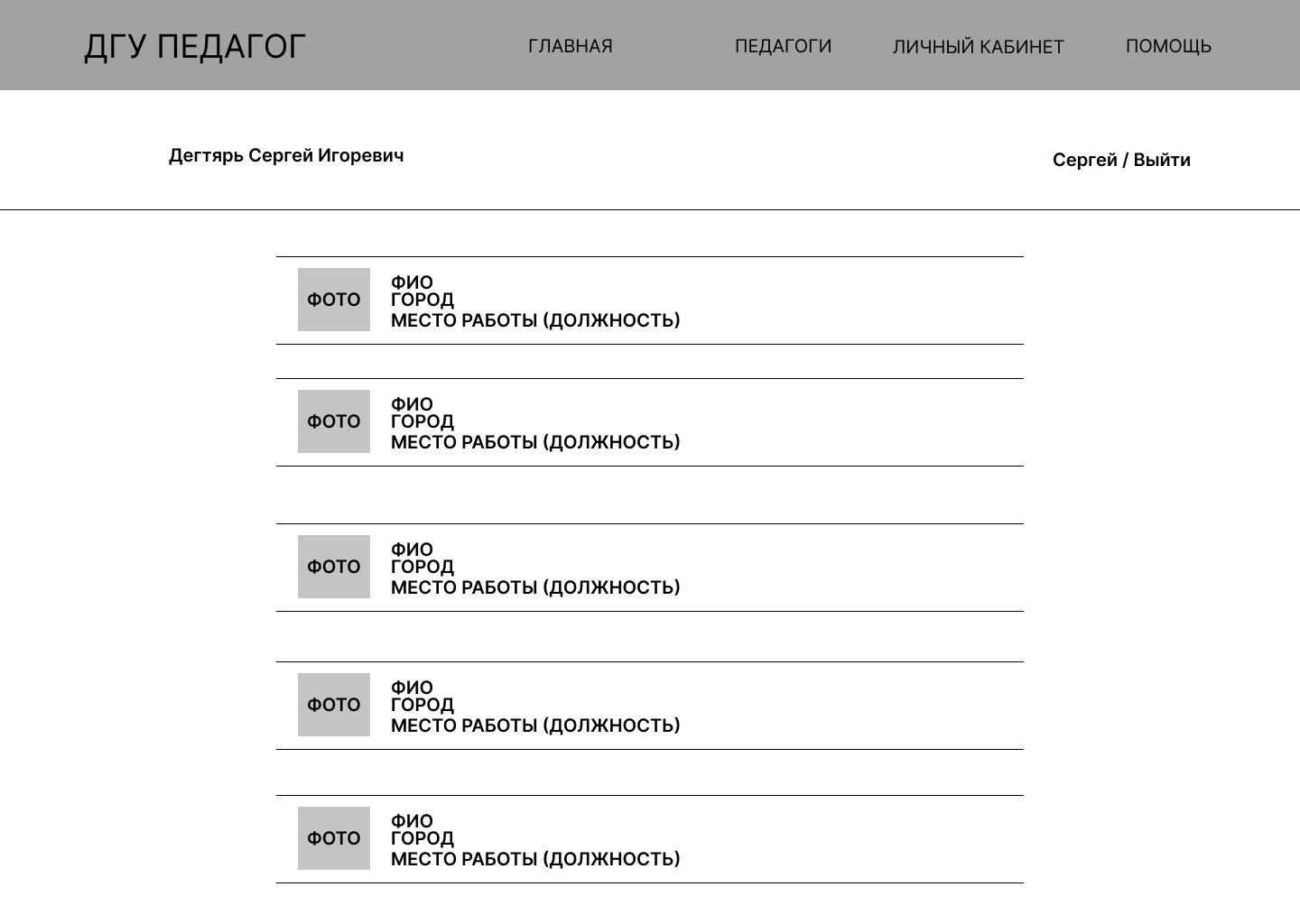
****

Рисунок 4 Педагоги

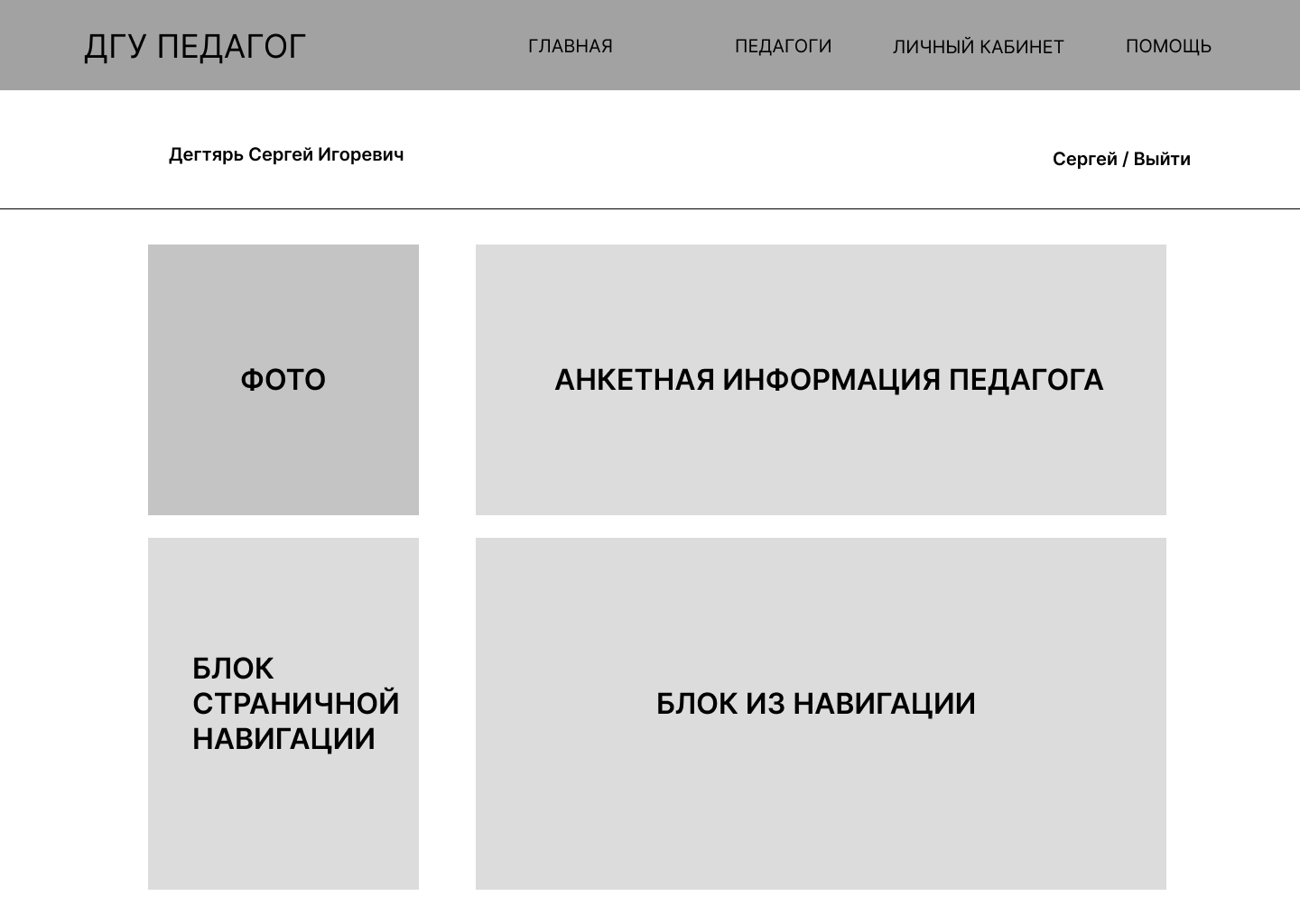
**СТРАНИЦА ПЕДАГОГА**

Рисунок 5 Страница педагога

**ЛИЧНЫЙ КАБИНЕТ**

* Выглядит альтернативно странице педагога, включая функцию редактирования профиля

## **3.2 Структурная схема**



Рисунок 6 Общая структура

# **Глава 2 «РАЗРАБОТКА WEB-САЙТА»**

## **4.1 Технология создания сайта**

На данный момент сайты есть уже практически у всех достаточно крупных компаний. А те, у кого сайта нет, мечтают его создать. И, в последнее время, большинство пользователей начали понимать, что создание сайта – не такое уж легкое дело. Существуют определенные **технологии создания сайтов**, которыми необходимо отлично владеть, чтобы создать хороший, работающий сайт. Ведь сайт является не только визиткой компании, но и одним (а зачастую основным) из рекламных, а также информационных инструментов. Поэтому встает вопрос, каким образом можно получить хороший сайт, который оправдает вложенные в него финансовые средства.

### ­4.1.1 PHP-скрипт

Большинство функциональных особенностей будет разрабатываться именно на этом языке.

Это скриптовый язык программирования, созданный для генерации HTML-страниц на веб-сервере и работы с базами данных. На данный момент он поддерживается практически всеми представителями хостинга, входит в «стандартный» набор для создания сайтов (LAMP – Linux, Apache, MySQL, PHP).

Благодаря своей простоте, скорости выполнения, богатой функциональности, распространению исходных кодов на основе лицензии PHP, этот язык является чуть ли не самым популярным в области **технологий создания сайтов**. Отличается наличием ядра и подключаемых модулей, «расширений»: для работы с базами данных, сокетами, динамической графикой, криптографическими библиотеками, документами формата PDF и т.п. Есть возможность разработать, а также подключить дополнительное расширение.

Возможности PHP очень обширны. Главным образом, PHP применяется при написании скриптов, работающих на стороне сервера; таким образом, PHP способен выполнять всё то, что выполняет любая другая программа CGI (например, обрабатывать данных форм, генерировать динамические страницы, отсылать и принимать cookies). Но PHP дает возможность выполнять также множество других задач.

Существуют три основных области, где используется PHP:

* создание скриптов для выполнения на стороне сервера;
* создание скриптов для выполнения в командной строке;
* создание приложений GUI, выполняющихся на стороне клиента.

Помимо этого, PHP:

* доступен для большинства операционных систем, включая Linux, многие модификации Unix (такие, как HP-UX, Solaris и OpenBSD), Microsoft Windows, Mac OS X, RISC OS, и многих других;
* включает поддержку большей части веб-серверов (для большинства серверов PHP поставляется в качестве модуля, для других, поддерживающих стандарт CGI, PHP может функционировать в качестве процессора CGI);
* поддерживает обширный круг баз данных;
* поддерживает DBX для работы на абстрактном уровне (таким образом можно работать с любой базой данных, использующих DBX); ODBC (т.е. вы можете работать с любой базой данных, поддерживающей этот стандарт);
* поддерживает "общение" с другими сервисами с использованием различных протоколов: LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (на платформах Windows), а также многих других;
* поддерживает стандарт обмена сложными структурами данных WDDX;
* поддерживает объекты Java, дает возможность использовать их в качестве объектов PHP;
* дает возможность формировать изображения, файлы PDF, ролики Flash, создаваемые "на лету"; способен выдавать любые текстовые данные (XHTML, другие XML-файлы); автоматически генерировать и сохранять в файловой системе вашего сервера;
* включает средства обработки текстовой информации, начиная с регулярных выражений Perl или POSIX Extended и заканчивая парсером документов XML;
* поддерживает многие другие расширения (функции поисковой машины mnoGoSearch, функции IRC Gateway, функции для работы со сжатыми файлами (gzip, bz2), функции календарных вычислений, функции перевода и др.)

### 4.1.2 JavaScript

Это пока еще относительно молодой язык программирования, но уже очень популярный в области **технологий создания сайтов**. На данный момент, работа над ним еще не закончена. Он постоянно дорабатывается и совершенствуется. Технический комитет работает над существенными расширениями, включая механизмы для сценариев, которые будут созданы для применения в Internet, а также более жесткой координацией с другими основными стандартами групп World Wide Web Консорциум и Wireless Application Protocol Форум. JavaScript уже сыграл очень важную роль в развитии технологий создания сайтов

В настоящее время используется третье издание ECMA-262, включающее мощные регулярные выражения, лучшую обработку строк, новые инструкции контроля, управления, перехват и обработку исключительных ситуаций, более жесткое определение ошибок, форматирование для числового вывода и незначительные изменения в ожидании ввода средств многоязычности и будущего развития языка.

### 4.1.3 HTML

Этот язык является базовым в области технологий создания сайтов, так как относительно легок в освоении. Но чрезмерная простота является и его недостатком. HTML (от английского Hyper Text Markup Language – язык разметки гипертекста) прекрасно отвечал требованиям раннего периода развития технологий создания сайтов, но с дальнейшим его развитием возникли существенные проблемы. HTML предоставляет следующие возможности:

* издавать сетевые документы с заголовками, текстом, таблицами, списками, фотографиями и т.п.;
* получать информацию из Сети через ссылки гипертекста при нажатии кнопки;
* создавать формы для посылки запросов на удаленные компьютеры, чтобы производить поиск информации, осуществлять бронирование, заказывать товары и т.п.;
* включать электронные таблицы, видео клипы, аудио клипы, и другие программные приложения непосредственно в их документы.

История разработки HTML довольно длительна. В каждой его версии разработчики пытались добиться того, чтобы HTML-страницы читались всеми браузерами, на всех компьютерных платформах.

Язык HTML предназначен для формирования и оформления страниц текста, который предназначен для просмотра в сети Интернет с помощью специальных программ – браузеров (основные из них Netscape Navigator (фирмы Netscape) и Internet Explorer (фирмы Microsoft).

В данном случае, под страницей подразумевается отдельный файл со специальными метками, которые указывают браузеру, как именно должен быть отображен на экране и в распечатке текст, содержащийся в этом файле. Помимо этого, могут еще присутствовать ссылки на файлы с графическими изображениями, которые будут отображаться вместе с текстом при просмотре.

Для формирования файла HTML нет необходимости в использовании мощного текстового процессора, более удобно использование, например, редактора Notepad (Блокнот). Содержимое HTML-страниц представлено в гипертекстовом виде, что предполагает наличие в документе, отображаемом на экране, выделенных определенным образом мест, щелкнув мышью по которым, можно переместиться в другую часть этой же страницы или на другую страницу.

### 4.1.4 СУБД и MySQL

SQL (от Structured Query Language – структурированный язык запросов) – создан для работы с реляционными базами данных. Он позволяет пользователям взаимодействовать с базами данных (просматривать, искать, добавлять, управлять данными). MySQL – многопользовательский, многопоточный сервер базы данных SQL. Имеет хорошую скорость и гибкость, если использовать его для хранения изображений и файлов. MySQL соответствует спецификации ANSI 92 SQL.

Его преимущества:

* поддержка нескольких одновременных запросов (многопоточность);
* возможность записи фиксированной, а также переменной длины;
* оптимизация связей с присоединением многих данных за один проход;
* гибкая система паролей и доступов;
* ODBC драйвер в комплекте с исходником;
* данные хранятся в формате ISO8859\_1;
* интерфейс с языками C и perl;
* легко управлять таблицей (включая добавление и удаление ключей и полей);
* до 16 ключей в таблице (в каждом ключе до 15 полей);
* поддержка ключевых полей, а также специальных полей в операторе CREATE;
* поддержка чисел длинной от 1 до 4 байт (ints, float, double, fixed), строк переменной длины и меток времени;
* утилита проверки и ремонта таблицы (isamchk);
* быстрая система памяти, основанная на потоках;
* при работе со строками регистр символов в обрабатываемых строках роли не играет;
* псевдонимы применимы не только к таблицам, но также к отдельным колонкам в таблице;
* все поля имеют значение по умолчанию. INSERT можно использовать на любом подмножестве полей.

Основные достоинства MySQL - скорость, устойчивость и легкость в использовании.

### 4.1.5 Bootstrap

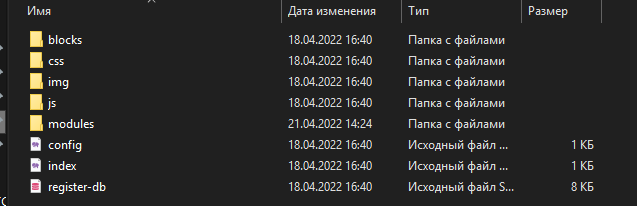
Технология Bootstrap — свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения. Наш сайт будет базироваться именно на этом решении, для максимально гибкого и упрощённого кодового смысла.

## 4.2 Начало разработки

### Начало разработки начинается с создание каталога проекта.

### 4.2.1 Корневая структура проекта

Всего в нашем проектном каталоге будет 5 основных папок.



1. BLOCKS
2. CSS
3. IMG
4. JS
5. MODULES
6. BLOCKS – это каталог который будет хранить в себе отдельные «почти-статичные» блоки со всего сайта, для примера любой сайт можно разбить на «шапку», «контент» и то что в самом низу «футер». Таким образом эти блоки будут подключаться однострочным кодом на всех будущих страницах сайта. Это очень удобно, учитывая что мы можем изменить текст в одном файле, но изменится текст на всех страничках, где есть этот блок.
7. CSS – это каталог в котором будут храниться файлы стилей нашего проекта. Так как в ходе разработки используется технология bootstrap, все основные компоненты будут лежать именно там.
8. Img – это каталог в котором будут храниться все изображения, графические решения, аватарки и прочее.
9. JS – это каталог в котором расположится основные JavaScript файлы. JavaScript поможет нам оживить наш сайт. На нём будут реализованы live\_search, а также навигационная панель в личном кабинете.
10. MODULES – основной каталог в котором будут храниться php файлы отвечающие за функциональные возможности нашего сайта. Формы регистрации, формы авторизации, хранение готовых запросов к СУБД и прочее.

|  |  |  |  |
| --- | --- | --- | --- |
| **Столбец** | **Тип** | **Null** | **По умолчанию** |
| ***id*** | int | Нет |  |
| login | varchar(11) | Нет |  |
| email | text | Нет |  |
| name | text | Нет |  |
| lastname | text | Нет |  |
| middlename | text | Нет |  |
| place\_born | text | Нет |  |
| speciality | text | Нет |  |
| place\_work | text | Нет |  |
| general\_experience | text | Нет |  |
| teacher\_experience | text | Нет |  |
| qualification\_category | text | Нет |  |
| institution\_address | text | Нет |  |
| institution\_phone | text | Нет |  |
| taught\_disciplines | text | Нет |  |
| additional\_load | text | Нет |  |
| about\_me | text | Нет |  |
| pass | varchar(32) | Нет |  |
| avatar | text | Нет | non-avatar.png |

### 4.2.2 Создание БД

**База данных register-db**

**Структура таблицы users**

Для администрирования базы данных и таблицы воспользуемся веб-приложением с открытым кодом «phpMyAdmin».

В рамках разработки создадим одну таблицу под названием “users”. Это таблица будет отвечать за сбор данных о пользователях (педагогах) сайта. Опишем структуру таблицы.

Основной первостепенной входящей информацией будут поля id\* email\* name\* lastname\* middlename\* pass\* и avatar\*. Заполнения этих полей будет происходить на этапе регистрации. Подчеркнём что поле \*avatar имеет значение по умолчанию как non-avatar.jpg, это значит что если пользователь не выбрал свою фотографию, то будет установлена аватарка по умолчанию.   
Заполнение оставшихся полей будет происходить непосредственно после регистрации, в личном кабинете, дабы уменьшить информационно-анкетную нагрузку на будущего пользователя. Обязательные поля указаны только при регистрации, это значит что на этапе регистрации нельзя проигнорировать какое-либо поле, но при редактировании в личном кабинете пользователь имеет право проигнорировать поля которые ему малоинтересны.

### 4.2.3 Создание страниц (PHP)

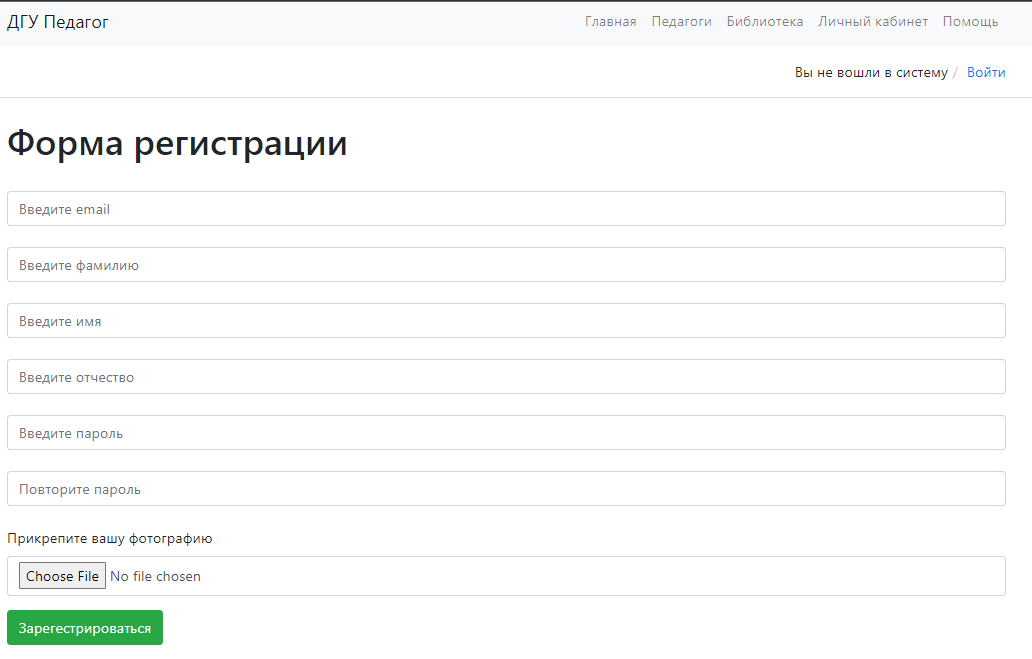


Рисунок 7 Форма регистрации

Опишем процесс работы формы регистрации «рисунок 6».

При нажатии кнопки «Зарегистрироваться», форма собирает введённые данные пользователем и отправляет их в обработчик auth.php. Обработчик принимает значения в формате utf-8, форматирует функцией filter\_var (то есть удаляет ненужные символы и пробелы) и заносит данные в таблицу users.

$email = filter\_var(trim($\_POST['email']),

FILTER\_SANITIZE\_STRING);

$name = filter\_var(trim($\_POST['name']),

FILTER\_SANITIZE\_STRING);

$lastname = filter\_var(trim($\_POST['lastname']),

FILTER\_SANITIZE\_STRING);

$middlename = filter\_var(trim($\_POST['middlename']),

FILTER\_SANITIZE\_STRING);

$pass = filter\_var(trim($\_POST['pass']),

FILTER\_SANITIZE\_STRING);

$repass = filter\_var(trim($\_POST['repass']),

FILTER\_SANITIZE\_STRING);

После регистрации сайт предлагает авторизоваться.

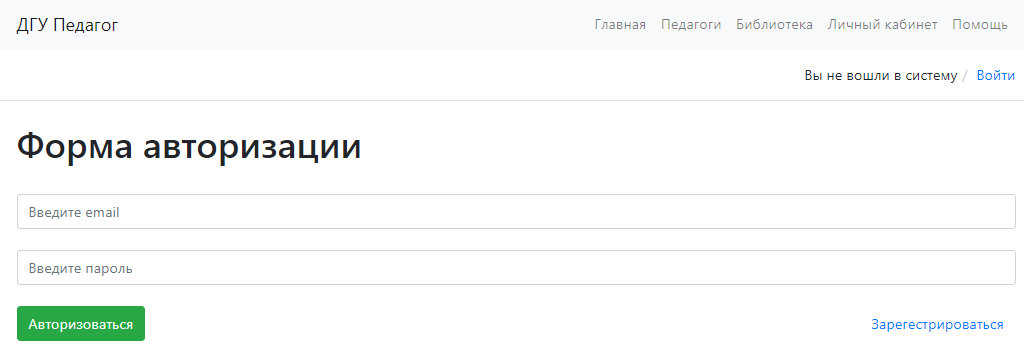


Рисунок 8 Форма авторизации

Запускаем сессию

session\_start();

Проверяем статус сессии. Если человек уже авторизован, происходит переадресация на главную страницу.

if (isset($\_SESSION['id']))

    {

        header("Location:".$siteurl);

    }

Иначе, проверяем данные которые вводит пользователь с данными из нашей таблицы.

else

    {

        if(isset($\_POST['submit']))

        {

            $email = filter\_var(trim($\_POST['email']), FILTER\_SANITIZE\_STRING);

            $pass = $\_POST['pass'];

            $query1 = mysqli\_query($db, "SELECT \* FROM `users` WHERE `email` = '$email' AND `pass` = '$pass'");

            if(mysqli\_num\_rows($query1)>0)

            {

                $user=mysqli\_fetch\_assoc($query1);

                $id=$user['id'];

                $name=$user['name'];

Если такие данные существуют, присваиваем сессии «имя пользователя» и «id» из таблицы ~users~.

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

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

                header("Location:/modules/user.php?id=".$\_SESSION['id']);

            }

После чего переадресуем в личный кабинет пользователя. Теперь пользователь авторизован.

Таким образом работает регистрация и авторизация на сайте. Всё достаточно просто.

### 4.2.4 Live\_Search (JavaScript)

Реализация поиска практически всегда приходится на язык JavaScript.

Начнём с поля вводимого текста. На языке html такое поле называется input. Такие же блоки используются в форме регистрации и авторизации.

<input type="search" class="form-control rounded" id="search" placeholder="Введите имя для поиска ..." aria-label="Search" aria-describedby="search-addon"/>

<button type="button" class="btn btn-outline-primary">Поиск</button>

Для работы с полями ввода есть набор событий:

1. focus - пользователь поставил фокус на поле ввода.

2. blur - пользователь убрал фокус с элемента вода.

3. input - пользователь как-то что-то ввел. Это лучше, чем keyup, ибо не надо проверки keyCode, второе потому что пользователь может в поле вставить текст, что тогда будет с Вашим keyUp? Событие input срабатывает каждый раз, как пользователь сделал изменение в поле ввода.

4. change - пользователь ввел данные и убрал фокус с поля ввода, то есть это input + blur. Событие происходит не так часто, как input, но и не заставляет проверять value на изменение, как это придется делать при blur.

5. submit - срабатывает на всей форме, при попытке отправить данные, в данной задаче скорее не интересное событие.

Для начало найдём input в javascript по id и присвоим ему класс-событие «oninput». Этот класс срабатывает всякий раз когда пользователь что-то вводит или стирает. После чего занесём ведённые данные в переменную val.

document.querySelector('#search').oninput = function(){

    let val = this.value;

Открывается цикл который уже будет выполнять саму функцию. Если val не равно пустоте то ищем элементы из input по всей области. Если есть совпадение, для всех остальных блоков добавляется класс hide, который скрывает их из поля видимости.

if (val != ' ') {

        searchItems.forEach(function(elem){

            if (elem.innerText.search(val) == -1) {

                elem.classList.add('hide');

            }

Иначе если пользователь удалит введённую конструкцию, нам нужно вернуть все блоки.

else {

    elem.classList.remove('hide');

}

Таким образом у нас работает достаточно простой блок поиска по сайту.

# Руководство пользователя

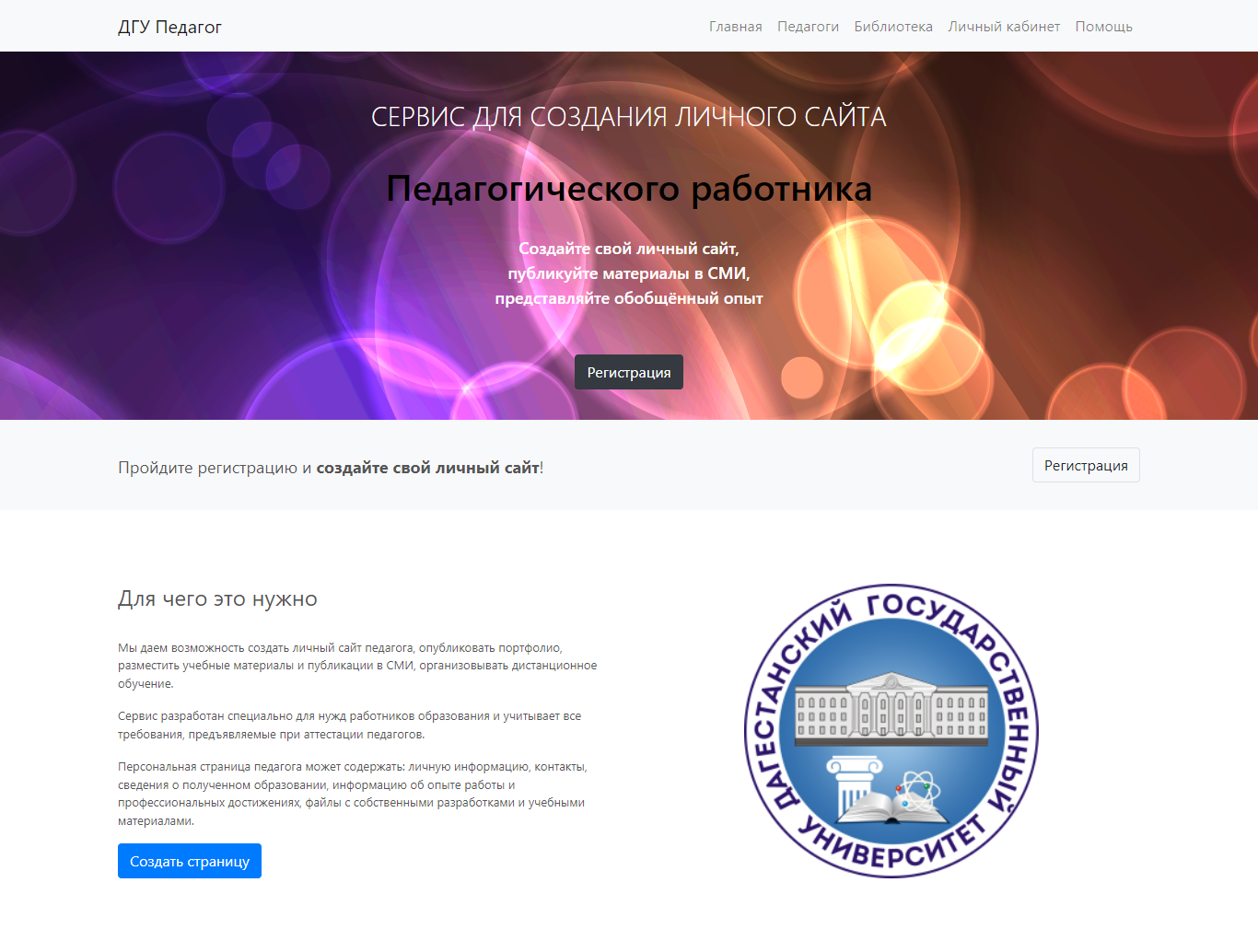


Рисунок 9 Главная страница

Попадая на главную страницу пользователь может зарегистрироваться по трём кнопкам на текущей странице.

Также пользователь может просмотреть по вкладке «библиотека» все доступные учебные материалы.

Нажатия на вкладку «педагоги» отобразит нам всех зарегистрированных педагогов с возможностью просмотра полной информации о них.

Нажатие на кнопку «личный кабинет» переадресует пользователя на страницу выбора либо регистрации, либо входа (условия статуса описаны в php конфигурации).

Вкладка «помощь» содержит ответы на часто задаваемые вопросы.

Вопросы делятся на технические и справочный каталог.

В технические вопросы можно отнести непредвиденные ошибки, возникшие входе эксплуатации веб-сервиса.

В справочный каталог относятся вопросы навигации сайта и также форма ­обратной связи.

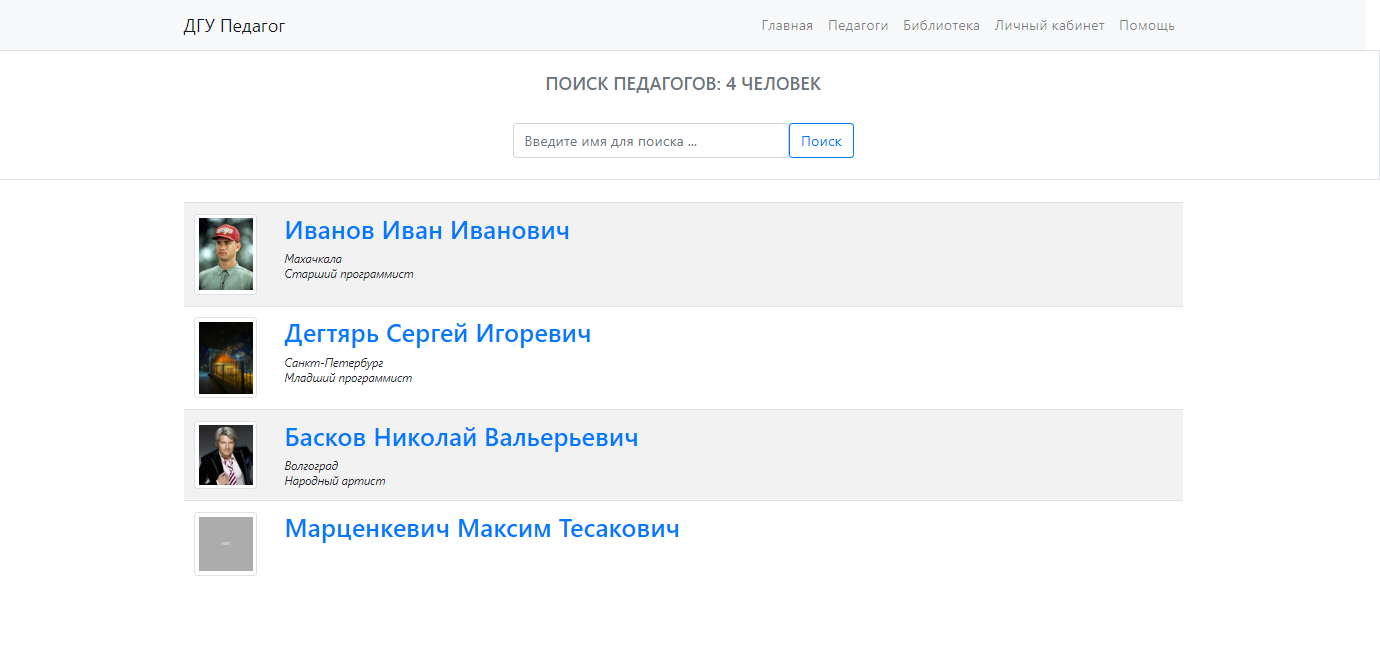


Рисунок 10 Педагоги

Страница педагоги отображает список зарегистрированных преподавателей. Для быстрого и удобного поиска нужного преподавателя используется строка поиск, реализованная на технологии Live\_Search (описанная на языке JavaScript).

Обновление списка происходит автоматически каждый раз, когда регистрируется новый пользователь.

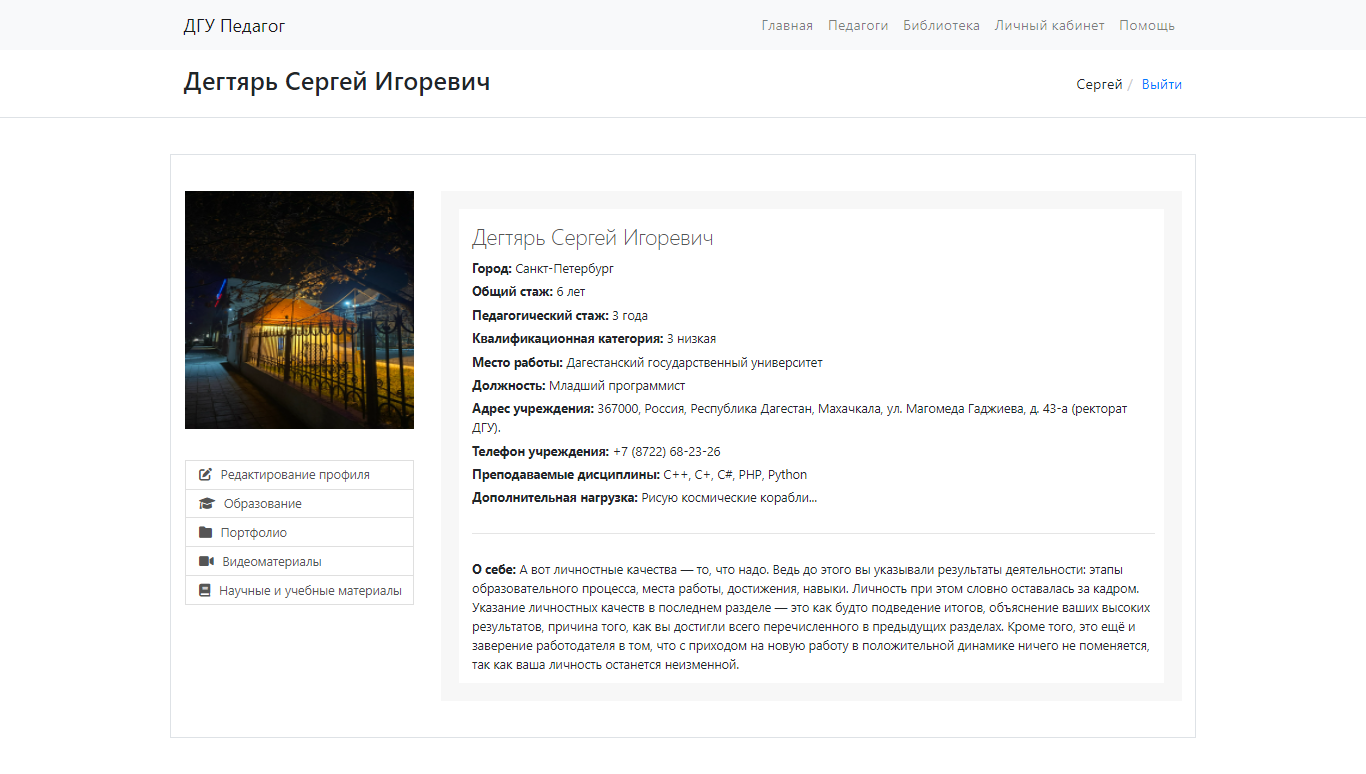


Рисунок 1 Личный кабинет

В личном кабинете отображается аватар и данные введённые на этапе регистрации. Под автаром расположен блок навигации. Он содержит:

1. Редактирование профиля (доступно только для личного профиля)
2. Образование - информация о профессиональном образовании, направлении подготовки и (или) специальности педагога. Возможность указания сведений о профессиональных переподготовках и повышениях квалификации.
3. Портфолио - публикация результатов профессиональной деятельности, результатов освоения обучающимися образовательных программ, дипломов и свидетельств за участия в конкурсах и олимпиадах и т.д.
4. Видеоматериалы - размещение ссылок на видеоматериалы, размещённые на сторонних сервисах: YouTube, Vimeo, Яндекс Диск, Облако Mail.ru и т.д. Загрузка видеофайлов на портал невозможна.
5. Научные и образовательные материалы - публикация образовательного контента для обучающихся: лекции, учебная литература, уроки, домашние задания и другие документы.

# ЗАКЛЮЧЕНИЕ

За время выполнения курсовой работы были изучены принципы разработки web-сайта, основы работы с СУБД MySQL, язык программирования PHP и система готовых шаблонов «Bootstrap»

На основе полученных знаний был спроектирован и разработан собственный web-сайт, который позволяет пользователю (педагогу) создать персональную страничку и наполнить её анкетной, карьерной и ресурсной информацией. С точки зрения информационной системы были организованы и автоматизированы основные потоки входной и выходной информации. Составлена графическая архитектура.

# СПИСОК ИСТОЧНИКОВ

1. <https://www.php.net/manual/ru/index.php>
2. <https://getbootstrap.com/docs/5.1/getting-started/introduction/>
3. <http://htmlbook.ru/>