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»

На тему: «Разработка 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

[ВВЕДЕНИЕ 3](#_Toc103823333)

[Глава 1. Анализ предметной области. 5](#_Toc103823334)

[1.1. Описание организации сайта педагога. 5](#_Toc103823335)

[1.1.1. Анализ деятельности организации. 5](#_Toc103823336)

[1.1.2. Анализ недостатков организации. 6](#_Toc103823337)

[1.2. Постановка задачи. 6](#_Toc103823338)

[1.2.1. Цель и задача проекта. 6](#_Toc103823339)

[1.2.2. Формирование требований к информационной системе. 7](#_Toc103823340)

[1.3. Планирование. 9](#_Toc103823341)

[1.3.1. Календарно-ресурсное планирование. 9](#_Toc103823342)

[1.3.2. Анализ бюджетных ограничений. 10](#_Toc103823343)

[1.3.3. Структурная схема. 11](#_Toc103823344)

[Глава 2. Разработка web-сайта. 12](#_Toc103823345)

[2.1. Технология создания сайта 12](#_Toc103823346)

[2.1.2 PHP-скрипт 12](#_Toc103823347)

[2.1.4. HTML 14](#_Toc103823348)

[2.1.5. СУБД и MySQL 16](#_Toc103823349)

[2.1.6. Bootstrap 17](#_Toc103823350)

[2.2. Начало разработки 18](#_Toc103823351)

[2.2.1. Эскизный проект 18](#_Toc103823352)

[2.2.2. Корневая структура проекта 21](#_Toc103823353)

[2.2.3. Создание базы данных. 22](#_Toc103823354)

[4.2.3 Создание страниц (PHP) 23](#_Toc103823355)

[4.2.4 Live\_Search (JavaScript) 25](#_Toc103823356)

[Руководство пользователя. 27](#_Toc103823357)

[Заключение. 30](#_Toc103823358)

[Список литературы. 31](#_Toc103823359)

# 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.

Курсовая работа посвящена изучению методов создания веб сайта для научного работника в формате контентных блоков а также реализации собственного веб-сайта, позволяющего автору своего профильного предмета, делиться большим спектром научных ресурсов с другими пользователями блога.

# Глава 1. Анализ предметной области.

1.1. Описание организации сайта педагога.

В настоящее время персональный сайт учителя не является редкостью, мы постепенно привыкаем к новому явлению. В сети Интернет можно найти множество персональных сайтов учителей-предметников, отличающихся по степени содержательности и структурированности. К тому же, те, кто столкнулся с аттестацией знают, что сейчас для аттестации, а также для стимулирующих выплат учителя предпочтительно иметь свой персональный сайт. Чаще всего учителя задают одни и те же вопросы: «Что такое сайт?», «В чем разница между сайтом и блогом?», «Достаточно ли для учителя иметь персональную страницу на сайте образовательного учреждения?», «Каким должен быть сайт учителя?», «Может ли персональный сайт учителя повлиять на качество образования?», «Нужна ли учителю специальная подготовка для создания сайта?». Прежде всего, будут рассмотрены вопросы структуры и содержания персонального сайта учителя.

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

Персональный сайт учителя-предметника – важнейший элемент учебно-воспитательного процесса, инструмент повышения качества образования, средство формирования информационно-коммуникативной культуры участников образовательного и воспитательного процессов. Создание персонального сайта позволит учителю презентовать свой педагогический опыт, получить навыки дистанционных форм обучения школьников, повысить уровень своей ИКТ-компетентности. Проект автоматизирует и структурирует объёмы информации, связанные с автором страницы, а именно: информацию об авторе, его личных ресурсах и также полную педагогическую информацию.

### 1.1.2. Анализ недостатков организации.

Новые способы деятельности сопровождаются не встречавшимися ранее проблемами. Для нивелирования рисков при реализации педагогического проекта в интернете и интернет – коммуникации необходимы новые компетенции, такие как защита персональных данных, соблюдение авторского права при интернет – публикации, защита личного интернет–пространства и информационно-образовательной среды. Основной проблемой является то, что личные права на произведение в педагогическом ресурсе могут быть нарушены путем присвоения авторства на произведение, неверного указания имени автора при размещении работы, внесения в нее правок без согласия создателя. Имущественные авторские права чаще всего нарушаются путем извлечения прибыли от использования творческой работы лицом, которое не имеет прав зарабатывать на произведении. Говоря простым языком, педагогу нужно направить экземпляр своего продукта компании, которая признает факт авторства, до того как он окажется в нерегулируемой информационной среде образовательного ресурса.

## 1.2. Постановка задачи.

### 1.2.1. Цель и задача проекта.

Создание собственного сайта помогает любому педагогу получить более высокий статус среди других работников образования. Это происходит из-за возможности поделиться своим опытом работы с сообществом российских и зарубежных учителей, со своими учениками, родителями и коллегами.

**Цель создания сайта:** создание информационной площадки для учащихся и преподающих, а также для демонстрации и представления собственного опыта, материал которого может быть полезным для других коллег.

**Задачи:**

1. Обеспечение учеников и коллег новой полезной информацией

2. Дистанционная поддержка учебного и воспитательного процесса.

3. Обеспечение взаимодействия учителя с учащимися, со службами сопровождения, с педагогами дополнительного образования.

### 1.2.2. Формирование требований к информационной системе.

Для того, чтобы сайт приводил потенциальных клиентов из поисковых систем, они должны его "видеть". Иными словами все страницы сайта должны быть проиндексированы, то есть внесены в память поисковых систем.

Предполагаются следующие разделы и типовые страницы:

* Главная страница.
* Страница педагогов.
* Страница библиотека.
* Личный кабинет.
* Помощь.

Главная страница:

* Яркий заголовок.
* Вводное пояснение.
* Блоки с регистрацией и «Создать страницу».

Страница педагогов:

* Отображение всех страниц педагогов в существующей системе.
* Быстрый поиск на технологии Live\_Search.

Страница библиотека:

* Отображение всех существующих файлов на сайте.
* Быстрый поиск на технологии Live\_Search.

Личный кабинет:

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

Помощь:

* Технические вопросы.

Регистрационная форма (\*-обязательно для заполнения):

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

Форма авторизации:

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

Форма «Забыли пароль» содержит поля:

* E-mail\* адрес пользователя, указанный при регистрации.

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:

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

## 1.3. Планирование.

### 1.3.1. Календарно-ресурсное планирование.

Анализ деятельности предприятия позволяет выделить ее сущности, определить первоначальные требования к функциональности и определить границы проекта. Модель предметной области должна быть документирована, храниться и поддерживаться в актуальном состоянии до этапа реализации. Создание организационной структура предприятия. Организационная структура — это один из способов наглядного представления структуры компании. Для создания органиграммы нужно собрать информацию о коллегах и определиться со способом построения. Формирование финансового плана предприятия. Разработка финансового плана предприятия – индивидуальный процесс для каждого отдельно взятого предприятия, зависящий от внутренних экономических особенностей и таланта специалистов финансового блока. Постановка задач. Формирование целей и задач предприятия (8 дней). Самый первый шаг при подготовке проекта в любой сфере: бизнесе, домашнем хозяйстве или образовании – состоит в определении целей и задач проекта. Этот шаг определяет, что мы хотим получить после завершения проекта, и какие действия нам необходимо предпринять для достижения этой цели. Создание структурной схемы web-сайта. Структурная схема сайта — это схема размещения его основных разделов и страниц относительно друг друга. Это план-схема, которая показывает, каким образом строится сайт, логическая связка его страниц. Разработка web-приложения Процесс разработки web-сайт — процесс, посредством которого потребности пользователей преобразуются в программный продукт. Загрузка web-сайт на хостинговый сервер. На данном этапе, уже готовый продукт (web-сайт), загружается на хостинг, что позволяет всем пользователям интернета пользоваться сервисом.

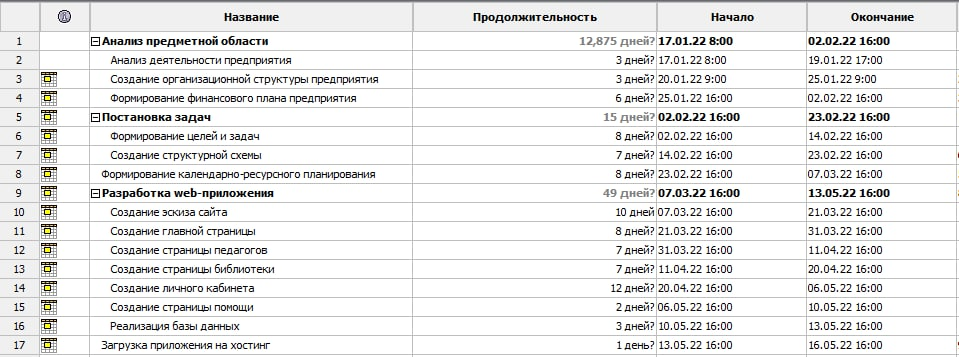


Рисунок 1 Календарно-ресурсный план

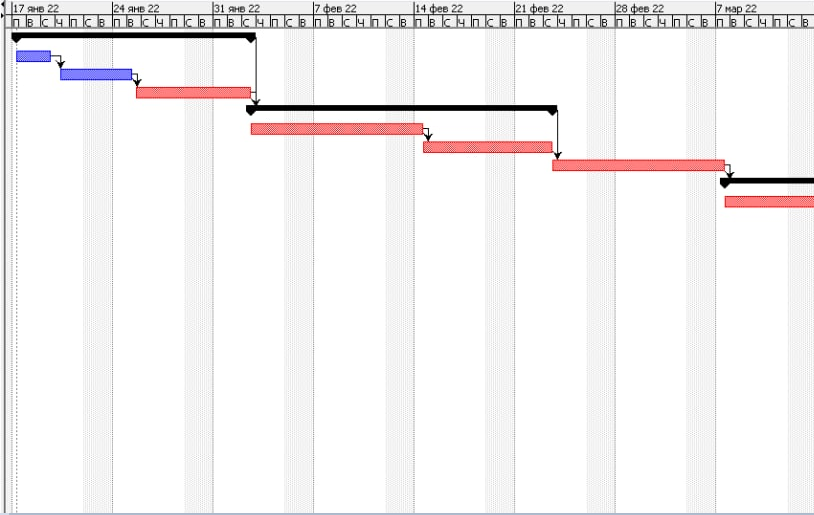


Рисунок 2 Диаграмма Ганта

### 1.3.2. Анализ бюджетных ограничений.

После определения основных этапов на стадии архитектуры проекта, с количеством ресурсов, 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 Yandex, установка на сервер, настройка оборудований, оплата хостинга и подключение доменного имени включая. Основным сервисным расходом можно считать продвижение сайта «SEO». 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.

### 1.3.3. Структурная схема.



Рисунок 3 Общая структура

Домашняя страница является основной точкой входа для нового пользователя. Данная схема подразумевает два логических маршрута.

Авторизованный маршрут – путь по которому может пройти user с наличием определённых прав доступа после того как представится системе.

Неавторизованный маршрут – путь по которому проходит user не представившейся системе.

# Глава 2. Разработка web-сайта.

## 2.1. 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.

### ­2.1.2 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.)

2.1.3. 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.

### 2.1.4. 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.

### 2.1.5. 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.

### 2.1.6. 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.

## 2.2. Начало разработки

### 2.2.1. Эскизный проект

После определения общей структуры проекта, количеством страниц и контентом, начинается этап разработки эскизного проекта как прототипа будущего сайта. Создание эскизного проекта будет осуществляться при помощи программы Figma. В силу функциональности некоторые детали будут использовать более «народную» программу Photoshop.

**Регистрация**

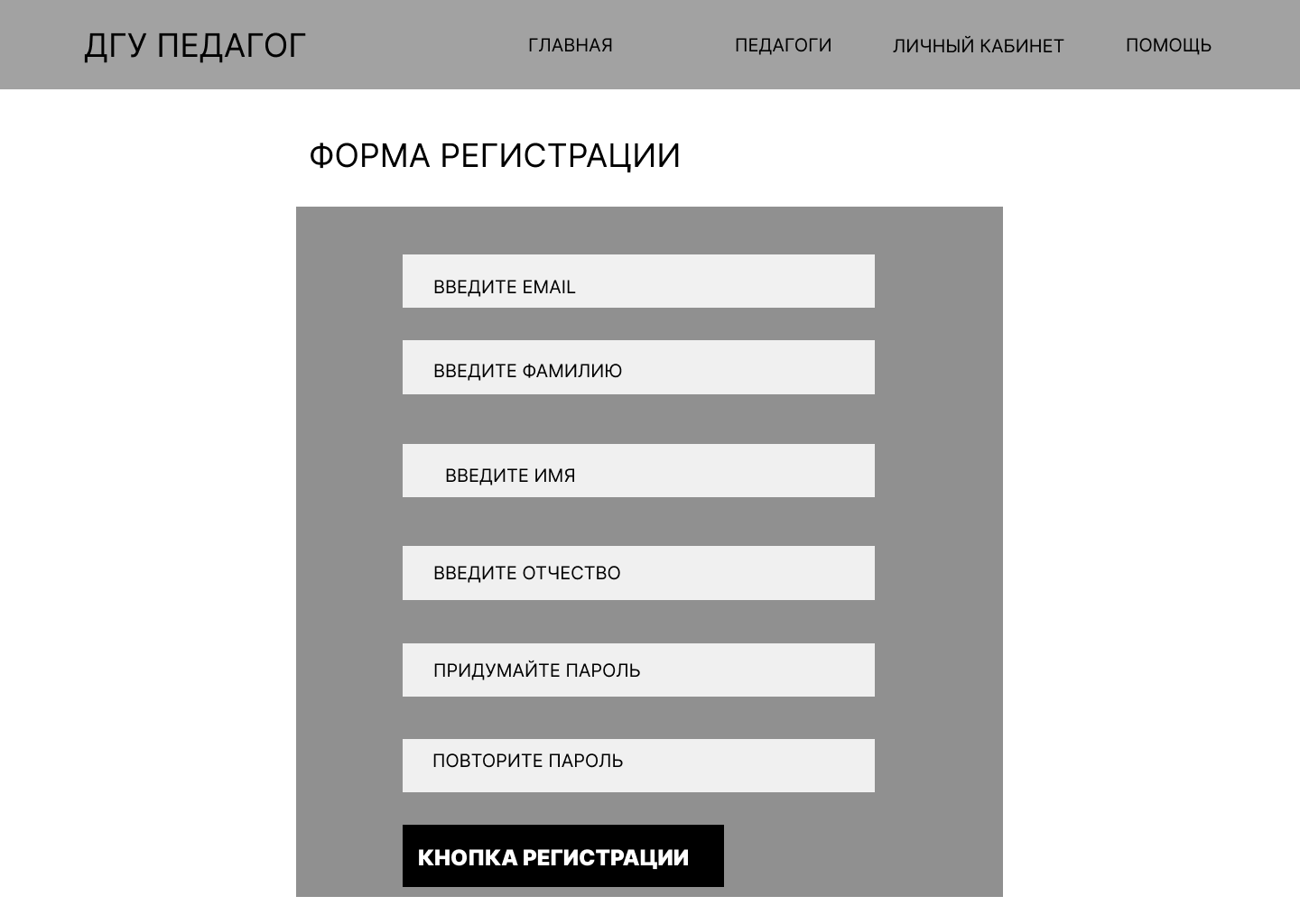


Рисунок 4 Регистрация

**Авторизация**

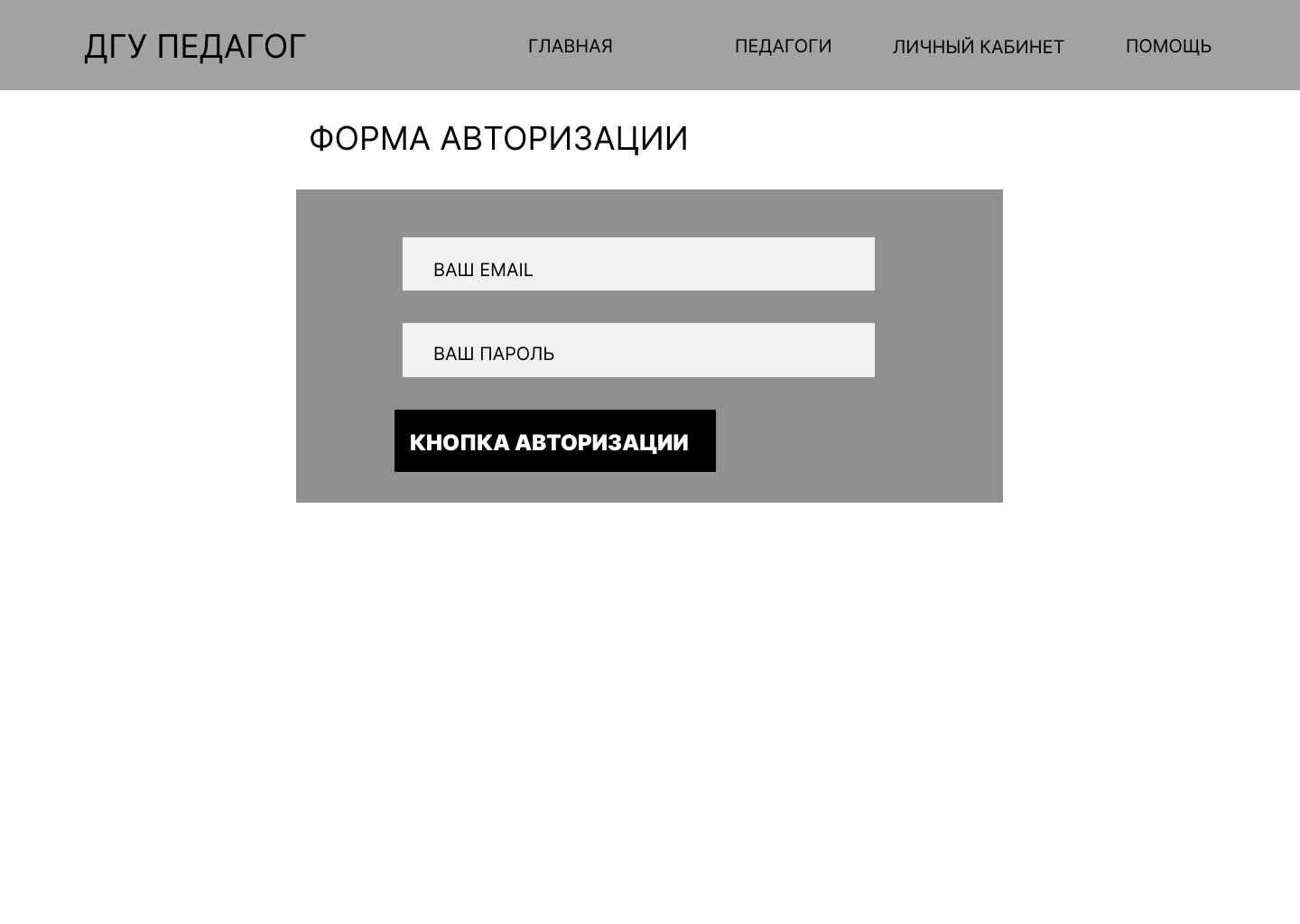


Рисунок 5 Авторизация

**Главная**

Рисунок 6 Главная страница

**Педагоги**

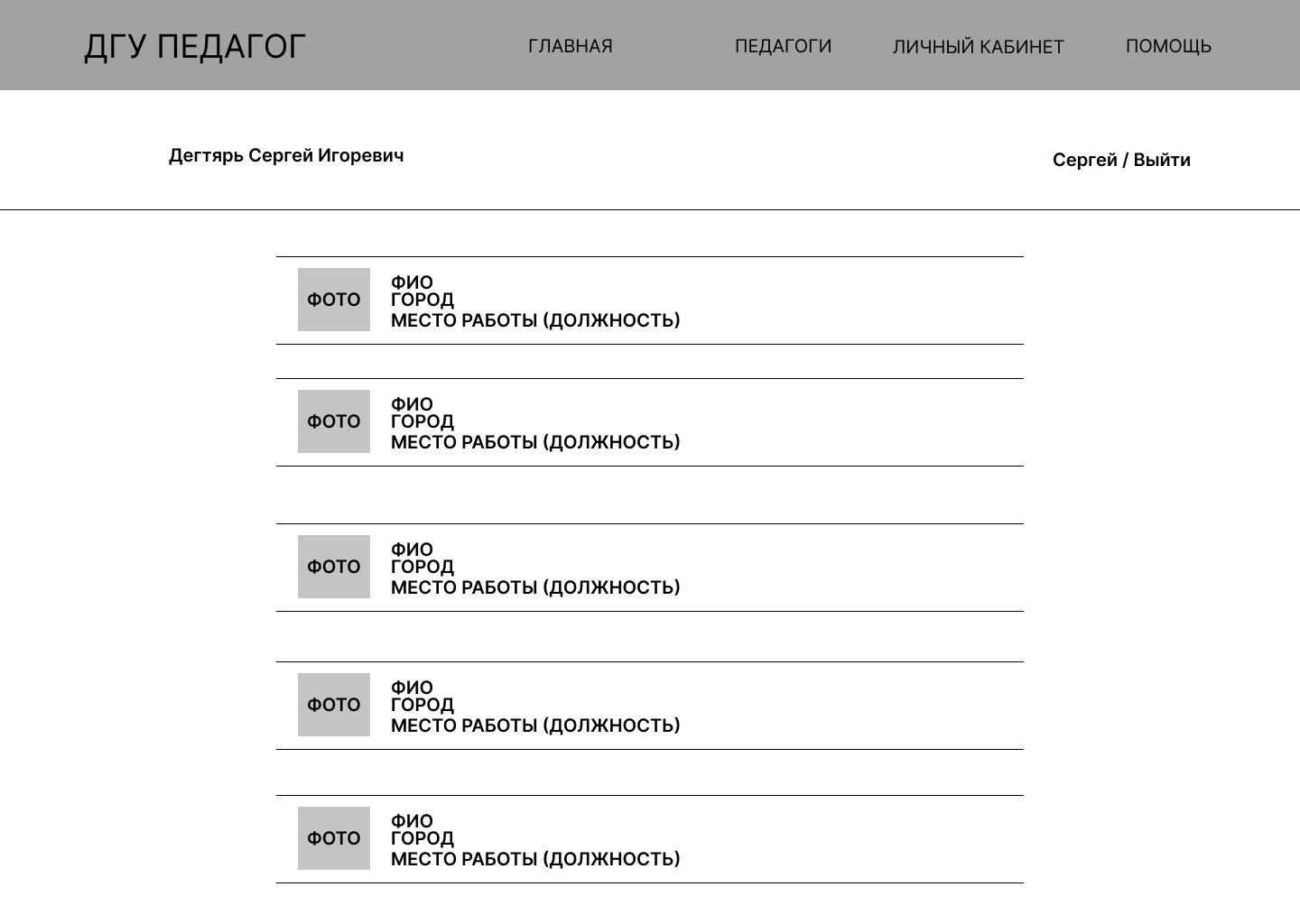
****

Рисунок 7 Педагоги

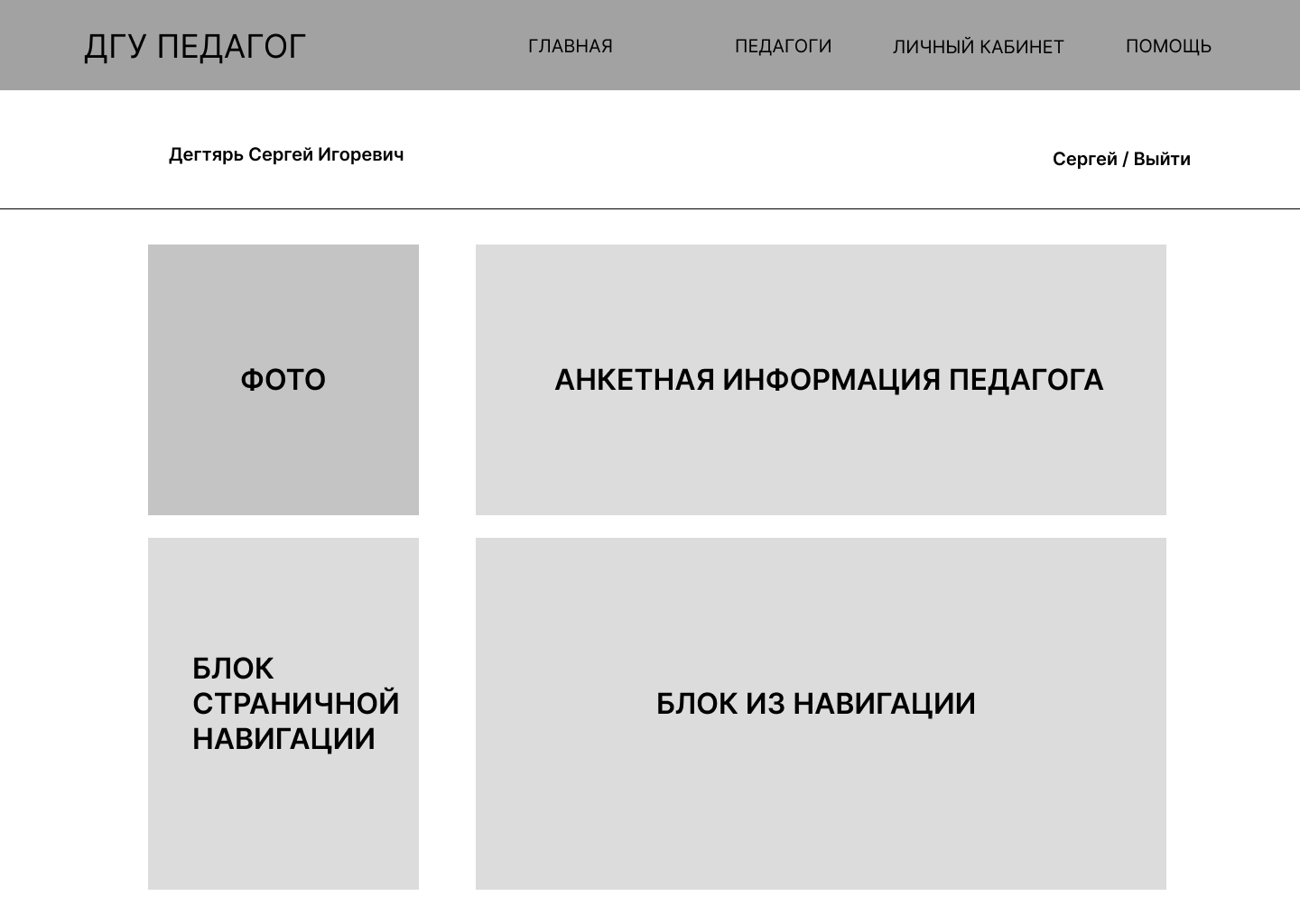
**Личная страница**

Рисунок 8 Страница педагога

### 2.2.2. Корневая структура проекта

Всего в нашем проектном каталоге будет 5 основных папок­:

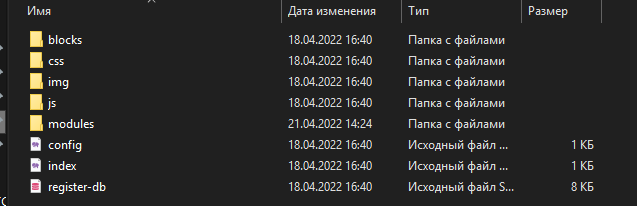


Рисунок 9 Структура каталога

1. 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.
2. 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.
3. Img – это каталог в котором будут храниться все изображения, графические решения, аватарки и прочее.
4. 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.
5. 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.

### 2.2.3. Создание базы данных.

**База данных register-db. Структура таблицы users.**

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

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)

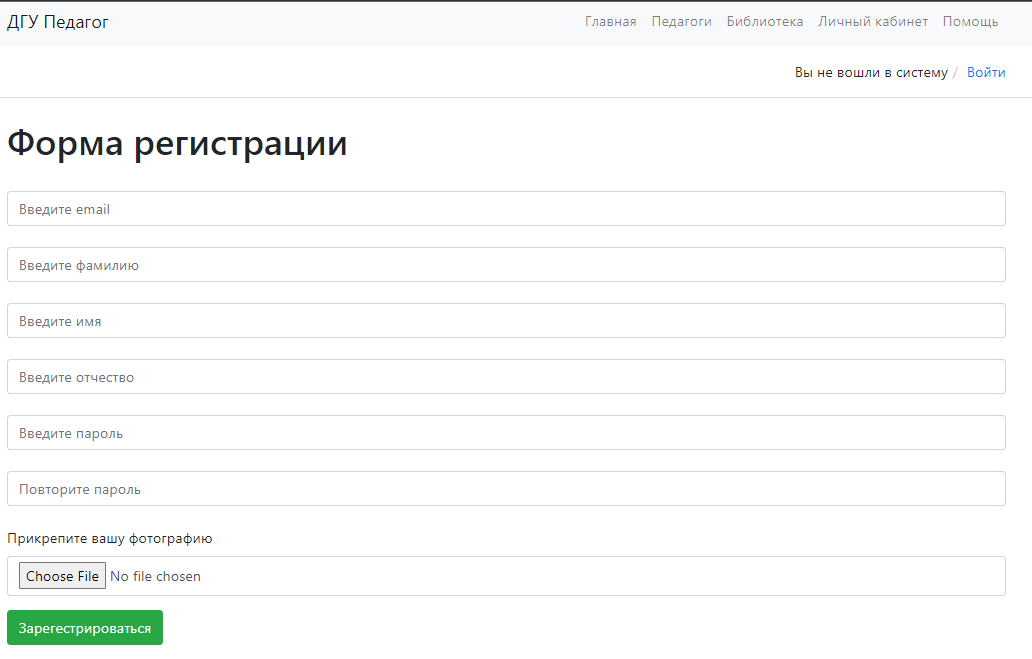
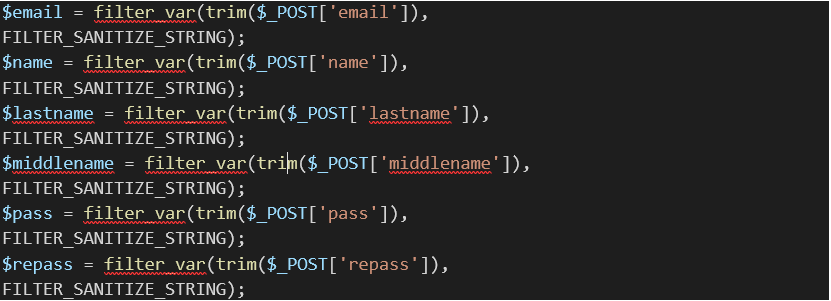


Рисунок 10 Форма регистрации

Опишем процесс работы формы регистрации «рисунок 10».

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.



После регистрации сайт предлагает авторизоваться.

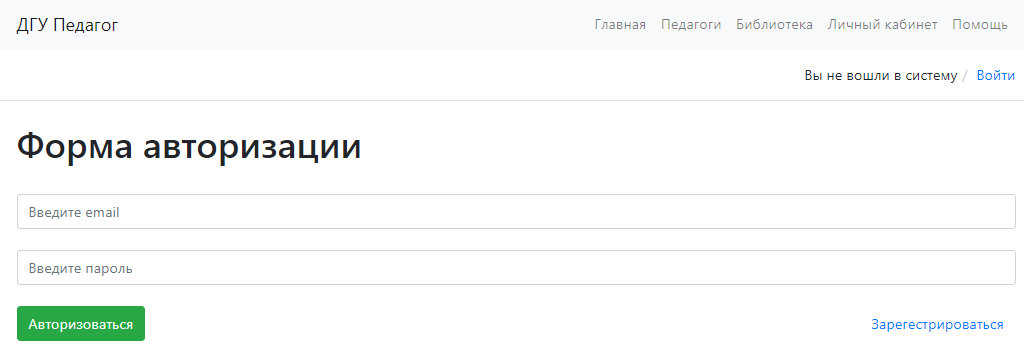
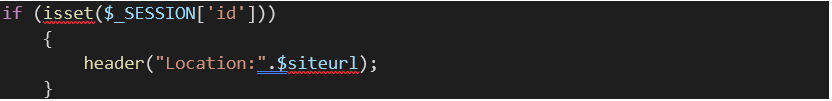


Рисунок 11 Форма авторизации

Запускаем сессию

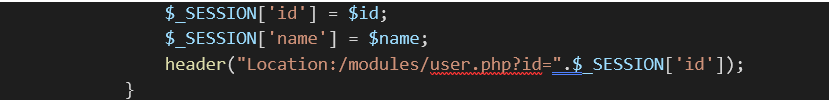


Проверяем статус сессии. Если человек уже авторизован, происходит переадресация на главную страницу.

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



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

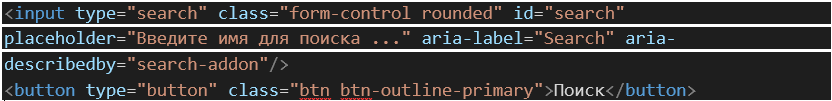
После чего переадресуем в личный кабинет пользователя. Теперь пользователь авторизован.

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.



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.

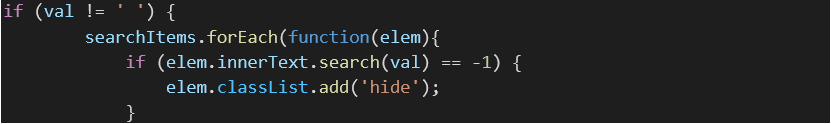
4. change - пользователь ввел данные и убрал фокус с поля ввода, то есть это input + blur. Событие происходит не так часто, как input, но и не заставляет проверять value на изменение, как это придется делать при blur.

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.



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.

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

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

# Руководство пользователя.

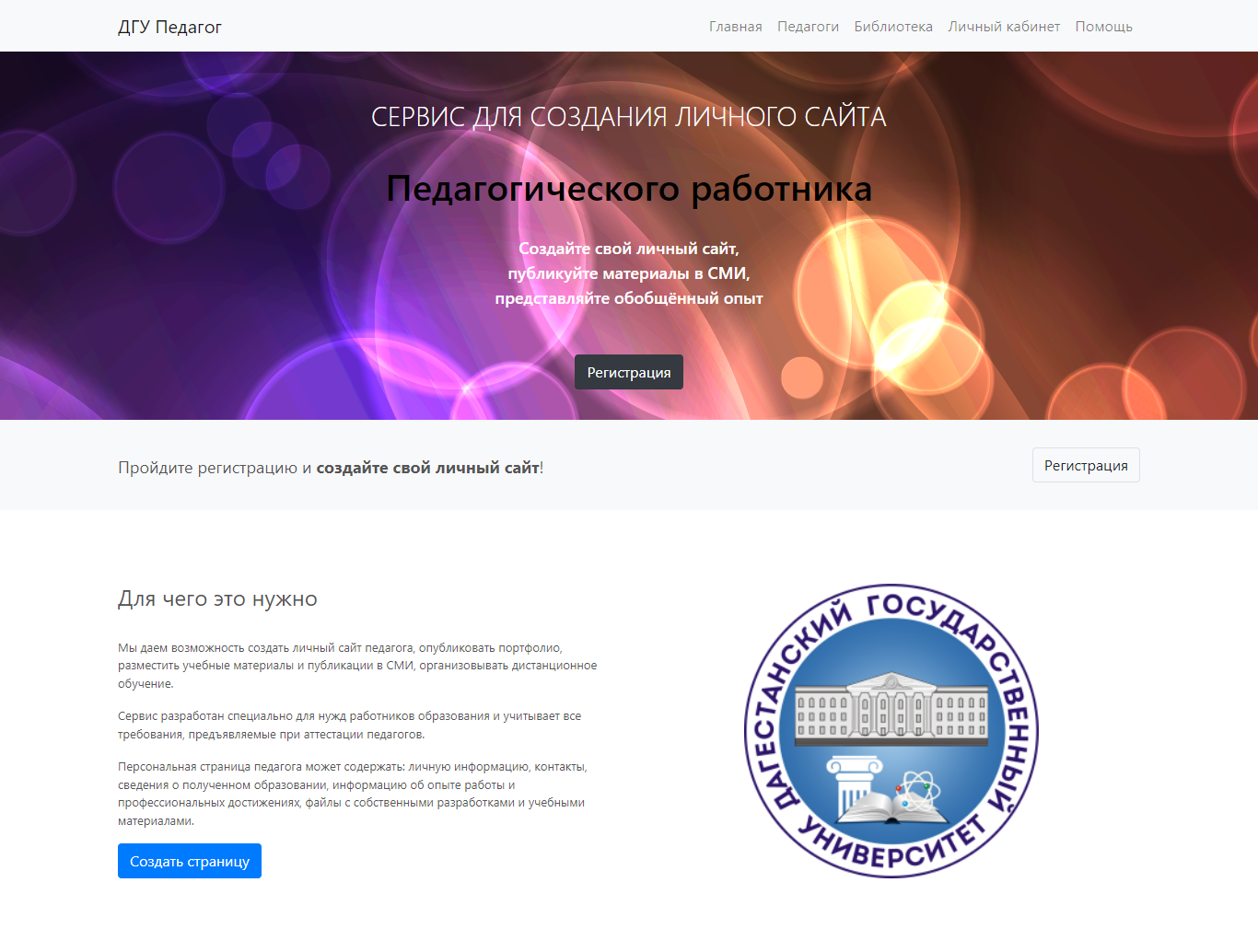


Рисунок 12 Главная страница

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.

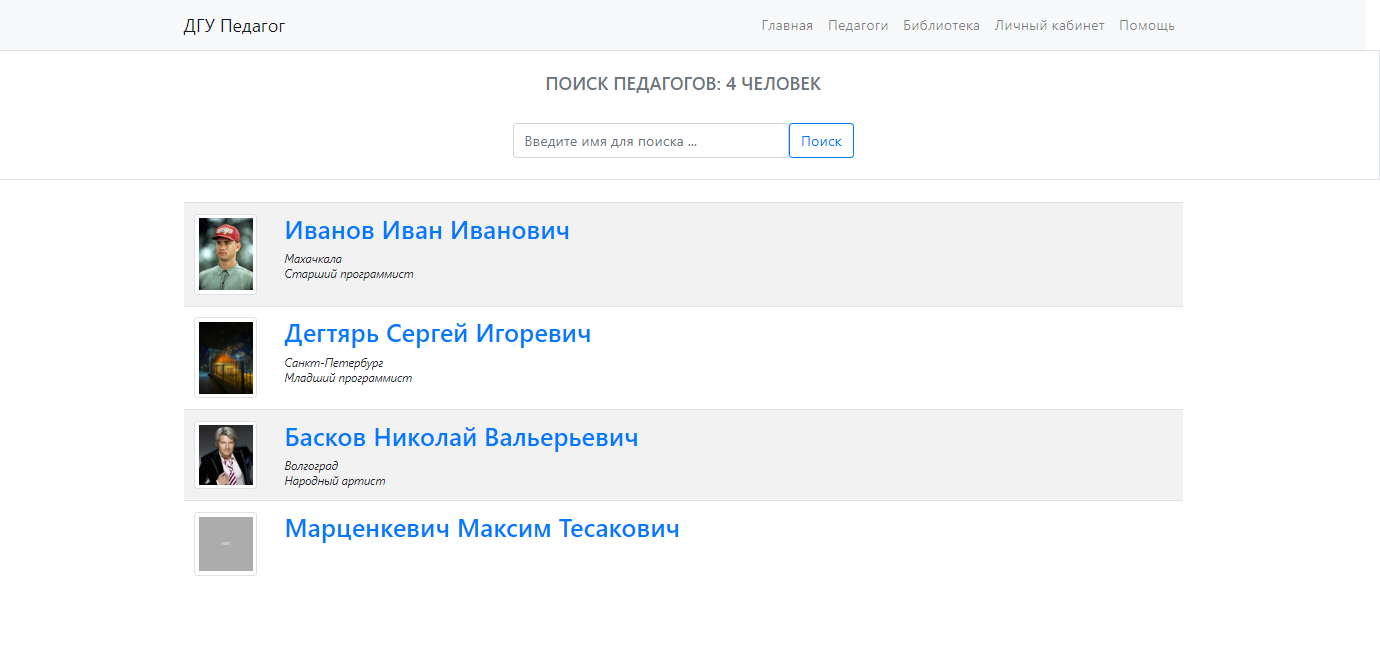


Рисунок 13 Педагоги

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.

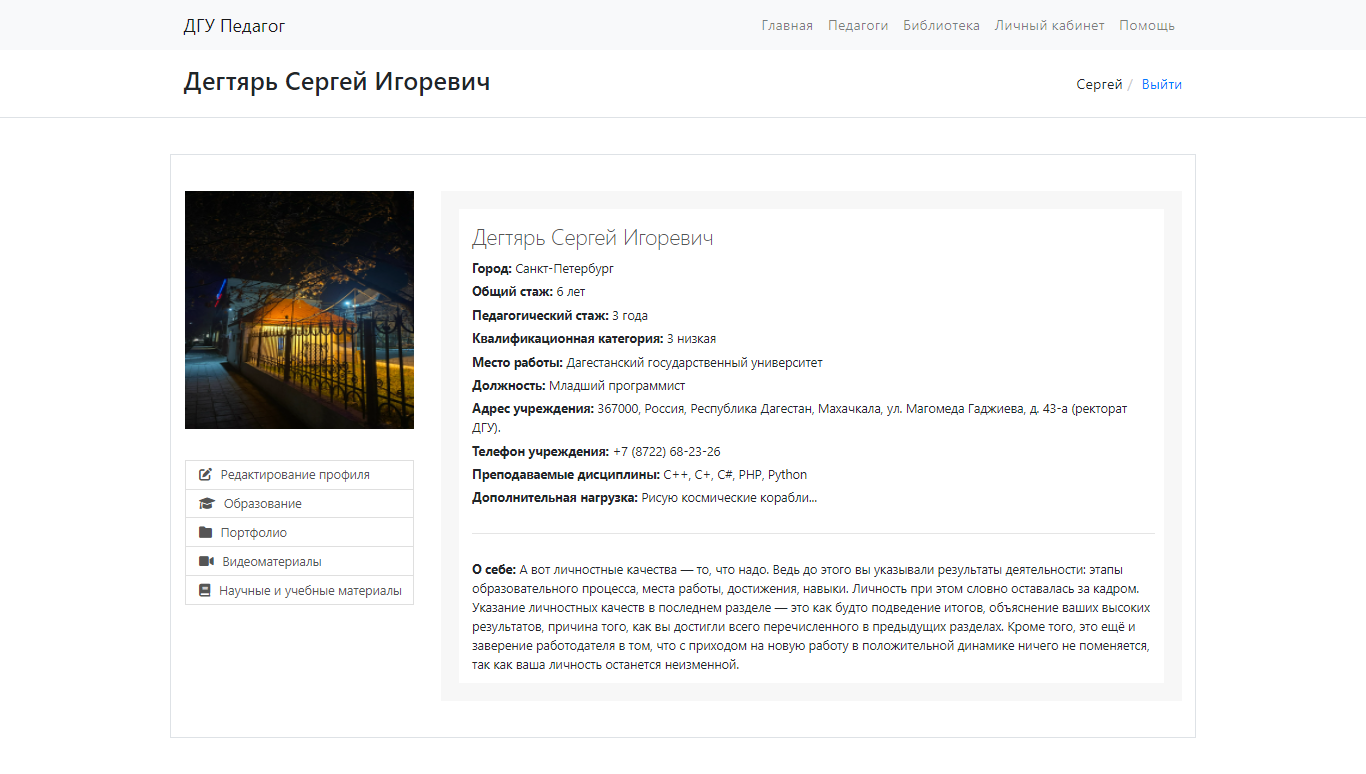


Рисунок 14 Личный кабинет

В личном кабинете отображается аватар и данные введённые на этапе регистрации. Под аватаром 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. Разработана лаконичная база данных.

# Список литературы.

1. ГОСТ 34.602-89. Техническое задание на создание автоматизированной системы.
2. ГОСТ 34.601-90. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания.
3. ГОСТ 34.201-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплексность и обозначение документов при создании автоматизированных систем;
4. РД 50-34.698-90. Методические указания. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов.
5. Никсон Робин. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. // Под ред. Вильчинский Н.Р., Черников С.В.: Питер, 2019.
6. Шилдт Г. Java 8. Полное руководство. – М.: ООО ― ИД Вильямс, 2015. – 1376 с.