|  |
| --- |
| РГРТУ  КУРСОВАЯ РАБОТА  735М – 09.04.02  Антонов А.А.  2018 |

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ АВТОМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В УПРАВЛЕНИИ

КАФЕДРА АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ

УТВЕРЖДЕН

КР –09.04.02–735М–02–18

КУРСОВАЯ РАБОТА

**«Разработка сайта-каталога»**

Пояснительная записка

КР–09.04.02–735М–02–18

Листов 37

Рязань 2018

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ АВТОМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В УПРАВЛЕНИИ

КАФЕДРА АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ

ЗАДАНИЕ

на курсовую работу по дисциплине

«Web-пpогpаммиpование»

Студент – Антонов Александр Анатольевич группа – 735М

1. Тема – «Разработка сайта-каталога»

2. Срок представления работы к защите – 25.12.2018 г.

3. Исходные данные для проектирования:

Язык программирования – PHP 5.5

СУБД – MySQL 5.6

Веб-сервер – Apache 2.4

4. Содержание курсовой работы

4.1. Задание.

4.2. Пояснительная записка, оформленная по ГОСТ 19.404-79.

Руководитель работы \_\_\_\_\_\_\_\_\_\_\_\_ 10.09.2018 г. Маркин А.В.

(подпись) (дата) (ФИО)

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_ 10.09.2018 г. Антонов А.А.

(подпись) (дата)

**Аннотация**

В курсовой работе осуществляется разработка сайта-каталога, представляющего информационный сайт Приокского завода цветных металлов состоящей из подсистем «Администратор» и «Пользователь».

В курсовой работе представлены:

* анализ предметной области;
* постановка задачи;
* структура базы данных;
* логическая структура системы;
* программная реализация сайта.

Программная реализация осуществлена на языке программирования PHP 5.5 с использованием СУБД – MySQL 5.6 и веб-сервера – Apache 2.4.

**Содержание**

[Введение 5](#_Toc533429392)

[1 Анализ предметной области 6](#_Toc533429393)

[2 Постановка задачи 7](#_Toc533429394)

[3 Описание средств разработки 8](#_Toc533429395)

[3.1 Язык разметки гипертекста XHTML 8](#_Toc533429396)

[3.2 Каскадные таблицы стилей CSS 9](#_Toc533429397)

[3.3 Web-сервер Apache 11](#_Toc533429398)

[3.4 Язык web-программирования PHP 13](#_Toc533429399)

[3.5 СУБД MySQL 15](#_Toc533429400)

[4 Разработка структуры базы данных 20](#_Toc533429401)

[5 Описание логической структуры сайта 23](#_Toc533429402)

[5.1 Подсистема «Пользователь» 25](#_Toc533429403)

[5.2 Подсистема «Администратор» 25](#_Toc533429404)

[6 Экспериментальная часть 26](#_Toc533429405)

[6.1 Функционирование подсистемы «Пользователь» 26](#_Toc533429406)

[6.2 Функционирование подсистемы «Администратор» 32](#_Toc533429407)

[Заключение 36](#_Toc533429408)

[Список использованных источников 37](#_Toc533429409)

[Приложение А 38](#_Toc533429410)

[А.1 Текст модулей сайта 42](#_Toc533429411)

[А.1.1 Файл INDEX.PHP 42](#_Toc533429412)

[А.1.2 Файл TOP.INC.PHP 45](#_Toc533429413)

[А.1.3 Файл BOTTOM.INC.PHP 45](#_Toc533429414)

[А.1.4 Файл LIB.INC.PHP 46](#_Toc533429415)

[А.1.5 Файл BASE\_REGISTRATION.PHP 49](#_Toc533429416)

[А.1.6 Файл AUTH.PHP 49](#_Toc533429417)

[А.1.7 Файл REGISTRATION.PHP 51](#_Toc533429418)

[А.1.8 Файл MENU.INC.PHP 52](#_Toc533429419)

[А.1.9 Файл PROFILE.PHP 53](#_Toc533429420)

[А.1.10 Файл CATALOG.PHP 54](#_Toc533429421)

[А.1.11 Файл ADD.PHP 56](#_Toc533429422)

[А.1.12 Файл EDIT.PHP 58](#_Toc533429423)

[А.1.13 Файл ITEM.PHP 60](#_Toc533429424)

[А.1.14 Файл USERLIST.PHP 60](#_Toc533429425)

[А.1.15 Файл USERCATALOG.PHP 62](#_Toc533429426)

[А.1.16 Файл STYLE.CSS 64](#_Toc533429427)

[А.2 Инструкция по развертыванию 66](#_Toc533429428)

# Введение

Веб-сайт в сети (как правило, Интернет) - это ресурс, который состоит из одной или нескольких веб-страниц с гипер-текстовым, текстовым, медийным (картинки, видео, аудио) и другими типами содержания, доступного по одному из протоколов веб-семейства.

Обычно сайт в Интернете представляет собой массив связанных данных, имеющий уникальный адрес и воспринимаемый пользователем как единое целое. Веб-сайты называются так, потому что доступ к ним происходит по протоколу HTTP.

Веб-сайт, как система электронных документов (файлов данных и кода) может принадлежать частному лицу или организации и быть доступным в компьютерной сети под общим доменным именем и IP-адресом или локально на одном компьютере.

Все сайты в совокупности составляют Всемирную паутину, где коммуникация (паутина) объединяет сегменты информации мирового сообщества в единое целое — базу данных и коммуникации планетарного масштаба [1].

В курсовой работе стоит задача разработки информационного интернет-ресурса, содержащий информацию о Приокском заводе цветных металлов, содержащий каталог выпускаемых драгоценных изделий.

# 1 Анализ предметной области

Добыча драгоценных металлов представляет собой процесс их извлечения из различного рода месторождений, в результате которого добываются концентраты и другие полупродукты, в составе которых присутствуют драгоценные металлы.

Производство драгоценных металлов определяется российским законодателем как процесс извлечения драгметаллов из уже добытых комплексных руд, концентратов и других полупродуктов, в составе которых присутствуют драгоценные металлы. Процесс производства драгметаллов также может быть связан с извлечением драгметаллов из лома и отходов, а также с аффинажем [2].

Сегодня представлено огромное количество типов изделий из драгоценных металлов. А если учесть, что каждое изделие имеет свои характеристики, то выбор становится очень громоздким и сложным. Цель данной курсовой работы– упростить эту задачу.

Основной задачей с точки зрения информационного наполнения сайта является обеспечение высокой концентрации информации, оперативности и объективности данных.

# 2 Постановка задачи

Цель курсовой работы - создание информационного сайта Приокского завода цветных металлов, хранящего и упорядочивающего информацию о различных видах изделий из драгоценных металлов. В конечном итоге сайт должен содержать исчерпывающую информацию обо всей продукции, продающейся на Приокском заводе цветных металлов. Каждое изделие должено характеризоваться описанием и фотографией, хранящимися в базе данных.

Сайт должен состоять из двух подсистем - «Администратор» и «Пользователь». Каждая из подсистем должна обладать соответствующим функционалом и возможностями.

1. Подсистема «Пользователь» должна выполнять следующие функции:
2. просмотр каталога;
3. сортировка записей каталога;
4. поиск записей в каталоге;
5. просмотр списка записей пользователя;
6. добавление записей в список пользователя;
7. удаление записей из списка пользователя.
8. Подсистема «Администратор» помимо вышеперечисленных функций должна выполнять следующие:
9. добавление записей в каталог;
10. редактирование записей каталога;
11. удаление записей из каталога;
12. просмотр списка всех пользователей.

Пользователь не может добавлять/удалять/редактировать записи непосредственно в базе данных, этим занимается администратор. Пользователь может только добавить запись к себе в профиль и удалить ее из него. Для просмотра записей на сайте «Пользователь» должен пройти обязательную регистрацию (авторизацию).

# 3 Описание средств разработки

Для разработки курсовой работы использовались следующие языки и программные средства:

* язык разметки гипертекста XHTML;
* каскадные таблицы стилей CSS;
* web-сервер Apache;
* язык web-программирования PHP;
* СУБД MySQL.

## **3.1** **Язык разметки гипертекста XHTML**

Язык разметки гипертекста (от англ. HyperText Markup Language, сокращённо HTML) – это стандартный язык разметки веб-документов, который определяет, как и какие именно элементы должны будут располагаться на веб-странице.

В основном веб-страницы создаются при помощи языка HTML или XHTML.

XHTML (от англ. Extensible Hypertext Markup Language – расширяемый язык разметки гипертекста) – язык разметки веб-страниц, по возможностям сопоставимый с HTML, однако предъявляющий более строгие требования к синтаксису [3].

Как и HTML, XHTML является подмножеством языка SGML, однако XHTML, в отличие от предшественника, соответствует спецификации XML.

В отличие от HTML, XML позволяет создавать собственные теги и таким образом формировать собственную структуру документа. В противоположность XML, HTML гораздо более строго определенный язык разметки с ограниченным набором тегов. В любом случае, общий характер XML позволяет рассматривать HTML-документы как XML-документы с набором тегов для отображения в веб-браузерах. Однако, старые стандарты HTML не до конца совместимы с XML. Например, в HTML необязательно закрывать тег, то есть закрывающий тег можно опускать. Чтобы устранить разрыв между этими двумя языками разметки и был разработан XHTML. По существу, это обычный HTML, в который добавили синтаксические правила XML для создания well-formed документов [4].

Для создания структуры документа и хранения служебной информации в нем предусмотрено много тегов, которые охватывают все необходимые пункты построения документа.

HTML-документ содержит следующие компоненты (рисунок 1):

* строку объявления типа документа;
* заголовок;
* тело документа.

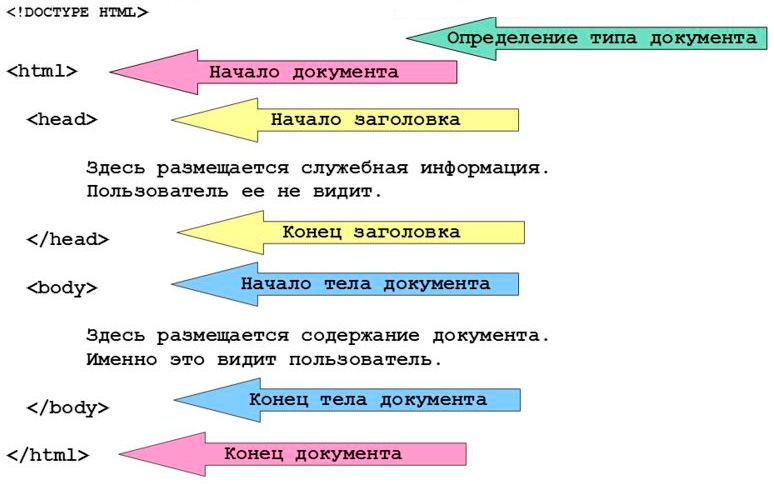


Рисунок 1 – Структура HTML-документа

## **3.2** **Каскадные таблицы стилей CSS**

CSS (от англ. Cascading Style Sheets – каскадные таблицы стилей) – технология описания внешнего вида документа, оформленного языком разметки.

Основным понятием CSS является стиль – т.е. набор правил оформления и форматирования, который может быть применен к различным элементам страницы [5].

Преимущественно используется как средство оформления веб-страниц в формате HTML и XHTML, но может применяться с любыми видами документов в формате XML, включая SVG и XUL.

Каскадные таблицы стилей используются создателями веб-страниц для задания цветов, шрифтов, расположения и других аспектов представления веб-документа. Основной целью разработки CSS являлось разделение содержимого (написанного на HTML или другом языке разметки) и оформления документа (написанного на CSS). Это разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода.

Каскадные таблицы стилей (Cascading Style Sheets, CSS) – это стандарт, определяющий представление данных в браузере. Если HTML предоставляет информацию о структуре документа, то таблицы стилей сообщают, как он должен выглядеть.

Таблицы стилей строятся в соответствии с определенным порядком (синтаксисом), в противном случае они не могут нормально работать. Таблицы стилей составляются из определенных частей (рисунок 2):



Рисунок 2 – Синтаксис описания стиля CSS

1. *Селектор* – это элемент, к которому будут применяться назначаемые стили. Это может быть тег, класс или идентификатор объекта гипертекстового документа.
2. *Свойство* определяет одну или несколько характеристик селектора. Свойства задают формат отображения селектора: отступы, шрифты, выравнивание, размеры и т.д.
3. *Значения* – это фактические числовые или строковые константы, определяющие свойство селектора.
4. *Описание* – совокупность свойств и их значений.
5. *Правило* – полное описание стиля (селектор + описание) [4].

Таким образом, таблица стилей — это набор правил, задающих значения свойств селекторов.

## **3.3 Web-сервер Apache**

Web-сервер – это сервер, являющийся местом хранения страниц веб-сайта вместе с базами данных и различными программными модулями сайта, это основополагающий элемент в работе всех web-ресурсов, хранящихся на нем. Но сама по себе ОС сервера не будет обеспечивать функционирование веб-сайта, необходимо определенное программное обеспечение, которым и является программный веб-сервер Apache.

Этот web-сервер был разработан в начале 90-ых годов и был предназначен для работы на платформе ОС Linux и Unix. Со временем, список поддерживаемых операционных систем значительно расширился. В настоящее время, **web-сервер Apache** функционирует на платформе ОC Windows, Mac OS, BSD, Linux, OS/2 и Novell NetWare. Веб-сервер имеет широкий набор расширений для работы с многими языками web-программирования:

* mod\_php для PHP;
* mod\_perl для Perl;
* mod\_wsgi, mod\_python для Python;
* apache-ruby для Ruby;
* apache-asp для ASP [6].

Дополняют функционал Apache многие разработчики по всему миру, но разработкой ядра веб-сервера занимаются только специалисты Apache Software Foundation. Например, русские специалисты занимаются разработкой расширения для легкой восприимчивости Apache к русскоязычной кодировке. Apache можно использовать как на web-сервере, работающем в глобальной сети, так и для локального использования, чтобы тестировать разрабатываемые веб-сайты.

Apache является самым распространенным веб-сервером, установленным на больше чем половине хостов по всему интернету. Главным образом, этому способствовало:

1. его бесплатная лицензия, которая позволяет работать с ним как новичкам, так и профессионалам web-индустрии;
2. кроссплатфоменность, которая является одним из главных плюсов по сравнению с вечным соперником – веб-сервером IIS;
3. открытость кода, благодаря которой многие специалисты могли дополнять и улучшать функционал Apache;
4. высокий уровень безопасности;
5. надежность и удобство работы.

Система конфигурации Apache основана на текстовых конфигурационных файлах. Имеет три условных уровня конфигурации:

* конфигурация сервера (httpd.conf);
* конфигурация виртуального хоста (httpd.conf c версии 2.2, extra/httpd-vhosts.conf);
* конфигурация уровня директории (.htaccess).

Имеет собственный язык конфигурационных файлов, основанный на блоках директив. Практически все параметры ядра могут быть изменены через конфигурационные файлы, вплоть до управления MPM. Большая часть модулей имеет собственные параметры.

Часть модулей использует в своей работе конфигурационные файлы операционной системы [6].

## **3.4** **Язык web-программирования PHP**

PHP – язык программирования, используемый на стороне WEB-сервера для динамической генерации HTML-страниц. Об этом говорит и расшифровка его названия: PHP - Personal HyperText Processor.

PHP – один из немногих языков программирования, созданных специально для разработки веб-приложений. Поэтому он включает в себя все функции, необходимые именно для работы на веб-сервере, и при этом лишен избыточности, свойственной многим его конкурентам [7].

Очень приятная особенность PHP – то, что его команды включаются в обычные HTML-страницы с помощью специальных тегов, которые и заставляют PHP-машину выполнять на сервере нужные действия. Программам на PHP не нужны специальные CGI-директории с особыми правами доступа. Более того, на одной страничке можно произвольно чередовать "простой" HTML и PHP-код.

PHP не зависит от платформы. PHP прекрасно интегрируется во все популярные веб-серверы: Apacce и IIS, Zens и Netscape Enterprise Server, работает под Windows и OS/2, MacOS и практически всеми UNIX-подобными системами. Как следствие – PHP работает практически у всех хостеров, разрешающих собственные выполняемые скрипты.

Замечательная особенность PHP – его интегрированность практически со всеми современными интернет-технологиями. PHP поддерживает большинство современных веб-протоколов: IMAP, FTP, POP, XML, SNMP и другие. PHP прекрасно работает с базами данных. Трудно найти СУБД, поддержка которой не была бы реализована в PHP: MySQL и MS SQL Server, PostgreSQL и Oracle, Sybase и Interbase... Один только перечень баз данных, поддерживаемых PHP, займет, наверное, целый экран.

PHP включает в себя огромное количество встроенных функций: обработки строк и массивов, работы с файловой системой и с HTTP, электронной почтой, датой и временем, кириллицей и другими национальными алфавитами... Благодаря им многие алгоритмы, требующие в большинстве языков написания программного кода размером в несколько экранов, реализуются на PHP одной командой (точнее, вызовом одной функции).

PHP – это скрипт-язык (scripting language), встраиваемый в HTML, который интерпретируется и выполняется на сервере [8].

Основное отличие от CGI-скриптов, написанных на других языках, типа Perl или C - это то, что в CGI-программах вы сами пишете выводимый HTML-код, а, используя PHP – вы встраиваете свою программу в готовую HTML-страницу, используя открывающий и закрывающий теги (<?php код ?>, <? код ?>, <?= код ?>) [7].

Вдобавок ко всему PHP понимает протоколы IMAP, SNMP, NNTP, POP3 и даже HTTP, а также имеет возможность работать с сокетами (sockets) и общаться по другим протоколам.

*Преимущества PHP:*

* легкий в освоении;
* развивается большим сообществом разработчиков;
* множество расширений и полезных библиотек;
* совместимость с популярными базами данных;
* поддерживается практически на любом сервере;
* кроссплатформенность (доступен под множество платформ);
* интеграция с HTML.

## **3.5 СУБД MySQL**

MySQL — это реля­ци­он­ная система управ­ле­ния базами дан­ных с откры­тым исход­ным кодом. В насто­я­щее время эта СУБД одна из наи­бо­лее попу­ляр­ных в веб-при­ло­же­ниях — подав­ля­ю­щее боль­шин­ство CMS исполь­зует именно MySQL (часто только её, без аль­тер­на­тив), а почти все веб-фрейм­ворки под­дер­жи­вают MySQL уже на уровне базо­вой кон­фи­гу­ра­ции (без допол­ни­тель­ных моду­лей).

MySQL поз­во­ляет хра­нить цело­чис­лен­ные зна­че­ния со зна­ком и без­зна­ко­вые, дли­ной в 1, 2, 3, 4 и 8 бай­тов, рабо­тает со стро­ко­выми и тек­сто­выми дан­ными фик­си­ро­ван­ной и пере­мен­ной дли­ны, поз­во­ляет осу­ществ­лять SQL-команды SELECT, DELETE, INSERT, REPLACE и UPDATE, обес­пе­чи­вает пол­ную под­держку опе­ра­то­ров и функ­ций в SELECT- и WHERE- частях запросов, рабо­тает с GROUP BY и ORDER BY, поддерживает групповые функ­ции COUNT, AVG, STD, SUM, MAX и MIN, поз­во­ляет использовать JOIN в запро­сах, в т.ч. LEFT OUTER JOIN и RIGHT OUTER JOIN, поддерживает репли­ка­цию, тран­зак­ции, работу с внеш­ними клю­чами и кас­кад­ные изме­не­ния на их осно­ве, а также обес­пе­чи­вает мно­гие дру­гие функ­ци­о­наль­ные воз­мож­но­сти.

Гиб­кость СУБД MySQL обес­пе­чи­вается под­держ­кой боль­шого коли­че­ства типов таб­лиц: поль­зо­ва­тели могут выбрать как таб­лицы типа MyISAM, под­дер­жи­ва­ю­щие пол­но­тек­сто­вый поиск, так и таб­лицы InnoDB, под­дер­жи­ва­ю­щие тран­зак­ции на уровне отдель­ных запи­сей. Есть и дру­гие типы таб­лиц, раз­ра­бо­тан­ные сооб­ще­ством.

СУБД MySQL появи­лась в 1995. Написана на C и C++, про­те­сти­ро­вана на мно­же­стве раз­лич­ных ком­пи­ля­то­ров и рабо­тает на раз­лич­ных платформах. C 2010 года раз­ра­ботку и под­держку MySQL осу­ществ­ляет кор­по­ра­ция Oracle. Продукт рас­про­стра­ня­ется как под GNU GPL, так и под соб­ствен­ной ком­мер­че­ской лицензией. Однако по усло­виям GPL, если какая-либопро­грамма вклю­чает исход­ные коды MySQL, то и эта про­грамма тоже должна рас­про­стра­няться по лицен­зии GPL. Для неже­ла­ю­щих откры­вать исход­ные тек­сты своих про­грамм как раз преду­смот­рена ком­мер­че­ская лицен­зия, кото­рая, в допол­не­ние к воз­мож­но­сти раз­ра­ботки под «закры­той» лицен­зи­ей, обес­пе­чи­вает каче­ствен­ную сер­вис­ную под­держку. Сообществом раз­ра­бот­чи­ков MySQL созданы раз­лич­ные ответв­ле­ния — Drizzle, OurDelta, Percona Server и [MariaDB](https://web-creator.ru/articles/mariadb), все эти ответв­ле­ния уже суще­ство­вали на момент полу­че­ния прав на MySQL кор­по­ра­цией Oracle.

Сей­час MySQL вместе с фор­ком MariaDB занимают почёт­ное пер­вое место, а сле­дом за ними идёт [PostgreSQL](https://web-creator.ru/articles/postgresql). Осталь­ные СУБД в веб-про­ек­тах исполь­зу­ются зна­чи­тельно реже [9].

Из преимуществ [СУБД](https://web-creator.ru/articles/about_databases) MySQL стоит отме­тить:

* Быстродействие. Благодаря внутреннему механизму многопоточности быстродействие MySQL весьма высоко.
* Безопасность. Довольно высокий уровень безопасности обеспечивается благодаря базе данных mysql, создающейся при установке пакета и содержащей пять таблиц. При помощи этих таблиц можно описать, какой пользователь из какого домена с какой таблицей может работать и какие команды он может применять. Пароли, хранящиеся в базе данных, можно зашифровать при помощи встроенной в MySQL функции password().
* Лицензия. Раньше лицензирование MySQL было немного запутанным; сейчас эта программа для некоммерческих целей распространяется бесплатно.
* Открытость кода. Благодаря этому вы сможете сами добавлять в пакет нужные функции, расширяя его функциональность так, как вам требуется.
* Надежность. Создатели MySQL потрудились на славу: насколько мне известно, этот пакет довольно стабилен и его трудно вывести из строя. Я не отслеживаю специально сводки результатов хакерских атак на MySQL, но мне ни разу не попадалось на глаза (в отличие от тех же Web-серверов) сообщение о том, что MySQL был поврежден в результате чьего-то злого умысла.
* Ресурсы. Это может зависеть от разных факторов, но в любом случае суперкомпьютер вам не потребуется.
* Сообщество. Как следствие открытости кода, бесплатности программы, стабильной и надежной ее работы образовалось сообщество людей, которые не просто лояльны к MySQL, но и всячески участвуют как в развитии самого пакета, так и в обучении менее опытных людей работе с ним. Существует огромное количество листов рассылки и конференций, где можно получить бесплатную помощь в любое время суток.
* Переносимость. В настоящее время существуют версии программы для большинства распространенных компьютерных платформ. Это говорит о том, что вам не навязывают определенную операционную систему. Вы сами можете выбрать, с чем работать, например с Linux или Windows, но даже в случае замены ОС вы не потеряете свои данные и вам даже не понадобятся дополнительные инструменты для их переноса.

Из недостатков [СУБД](https://web-creator.ru/articles/about_databases) MySQL стоит отме­тить:

* Транзакции - позволяют объединить несколько SQL-запросов в одну единицу работы и в случае сбоя любого из запросов, входящего в эту единицу, выполнить откат, чтобы вернуть данные в исходное состояние.
* Триггеры - служат для автоматизации контроля за состоянием и работой базы данных. Триггер хранится в базе и срабатывает, когда происходит определенное событие. Возьмем тот же пример с переводом денег: в случае сбоя при выполнении второго запроса сработает триггер, который выполнит откат либо пошлет сообщение администратору базы данных.
* Хранимые процедуры - это несколько SQL-команд, которые хранятся в базе данных под неким именем и в совокупности выполняют некую функцию. При помощи хранимых процедур можно расширить синтаксис SQL так, что он будет похож на обычный язык программирования (например, Oracle PL/SQL).
* Вложенные запросы - позволяют подставлять значения в условия отбора динамически, по результатам выполнения другого запроса. Если без всего вышеперечисленного еще можно как-то обойтись, то отсутствие вложенных запросов иногда очень портит жизнь. Например, чтобы узнать, какой автомобиль перевез грузов больше, чем в среднем по автопарку, нужно сделать такой SQL-запрос:

*SELECT auto FROM autopark WHERE massa > !Больше чего? Я понятия не имею, каково среднее значение!*

Для этого среднее значение в поле massa нужно вычислить:

*SELECT AVG(massa) FROM autopark*

Если поддерживаются вложенные запросы, то эти два запроса можно вложить друг в друга:

*SELECT auto FROM autopark WHERE massa >( SELECT AVG(massa)*

*FROM autopark)*

Но в случае с MySQL среднее значение приходится находить отдельно и подставлять в другой запрос непосредственно в CGI-сценарии, что, несомненно, сказывается на производительности.

* Инструкция UNION - попросту говоря, она объединяет вывод нескольких запросов в один, с возможностью исключить дубликаты строк.
* Каскадное обновление данных - позволяет удалять и обновлять связанные данные. Например, при удалении из базы данных записи о клиенте из связанных таблиц автоматически удаляются все записи о заказах этого клиента.

# 4 Разработка структуры базы данных

При разработке любого веб-приложения встает вопрос о необходимости хранения данных. В случае простых приложений, а также при относительно небольшом объеме данных и интенсивности запросов к ним последние можно хранить в обычных текстовых файлах или в сессиях. Однако если приложение предполагает наличие тысяч и более записей, высокую интенсивность запросов, защиту данных, выборку нужных сведений по различным критериям, а также их анализ, то целесообразнее использовать базы данных.

База данных представляет собой хранилище информации, представленное одним или несколькими файлами. В настоящее время наиболее широко используются так называемые реляционные базы данных, в которых информация организована в виде таблиц, состоящих из строк и столбцов. Таблицы могут быть связаны между собой, образую сложную систему. Создание баз данных и доступ к ним обеспечивает система управления базами данных (СУБД). Вместе с тем доступ к базам данных через СУБД возможен и из приложений, написанных на том или другом языке программирования.

PHP имеет наборы специальных функций для работы со многими СУБД, такими как Microsoft SQL Server (MS SQL), MySQL, Oracle, Interbase, Firebird, PostgreeSQL, Informix, Sybase, Ingres и IBM DB2. Кроме того, PHP соответствует стандарту ODBC (Open Database Connectivity – открытый интерфейс доступа к базам данных), что позволяет обращаться к таким СУБД, как MS Access. В курсовой работе используется СУБД MySQL.

Для выполнения поставленной задачи разработана следующая структура базы данных (рисунок 3). База данных включает в себя 3 таблицы:

* таблица пользователей (users) – содержит информацию о зарегистрированных пользователях сайта;
* таблица товаров (tovar) – содержит основную информацию о производимой продукции, продаваемой заводом, так же является базовой для вывода каталога;
* таблица соответствия записей таблицы товаров и таблицы пользователей (usercatalog) – необходима для формирования пользовательского каталога.

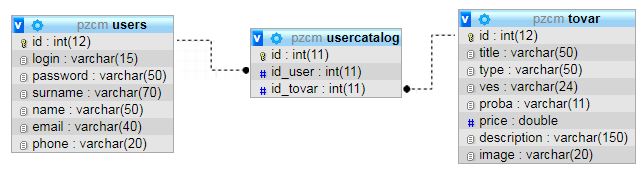


Рисунок 3 – Общая структура базы данных

Таблица USERS (пользователи) – содержит информацию о пользователях, зарегистрированных на данном сайте (таблица 1). Поле ID является первичным ключом таблицы (автоинкрементное поле). Поля LOGIN и PASSWORD содержат информацию, необходимую для авторизации пользователя на сайте. При этом в БД хранится не сам пароль, а его хэш-функция. Остальные поля содержат дополнительную информацию о пользовательском аккаунте.

Таблица 1 – Атрибуты таблицы USERS (пользователи)

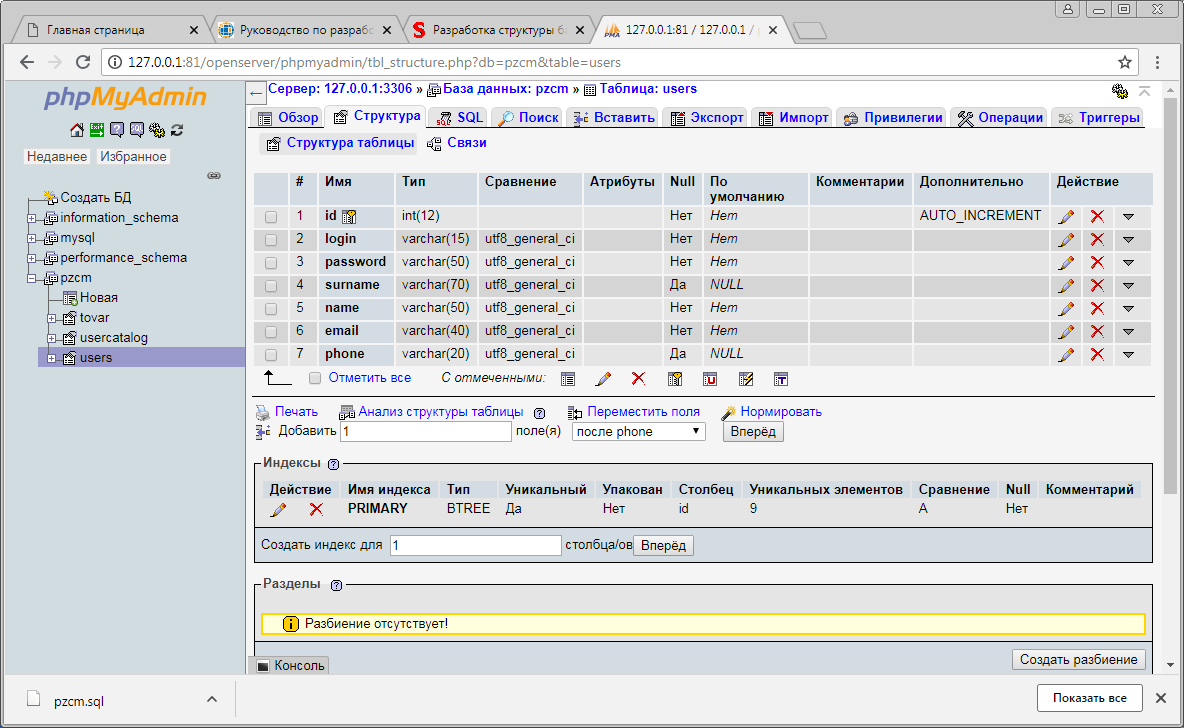


Таблица tovar (товары) – основная таблица каталога, содержащая информацию о видах изделий из драгоценных металлов, продаваемых Приокским заводом цветных металлов (таблица 2). Поле ID является первичным ключом таблицы (автоинкрементное поле). В таблице приведены основные характеристики изделий, их стоимость и фотография.

Таблица 2 – Атрибуты таблицы tovar (предлагаемые товары)

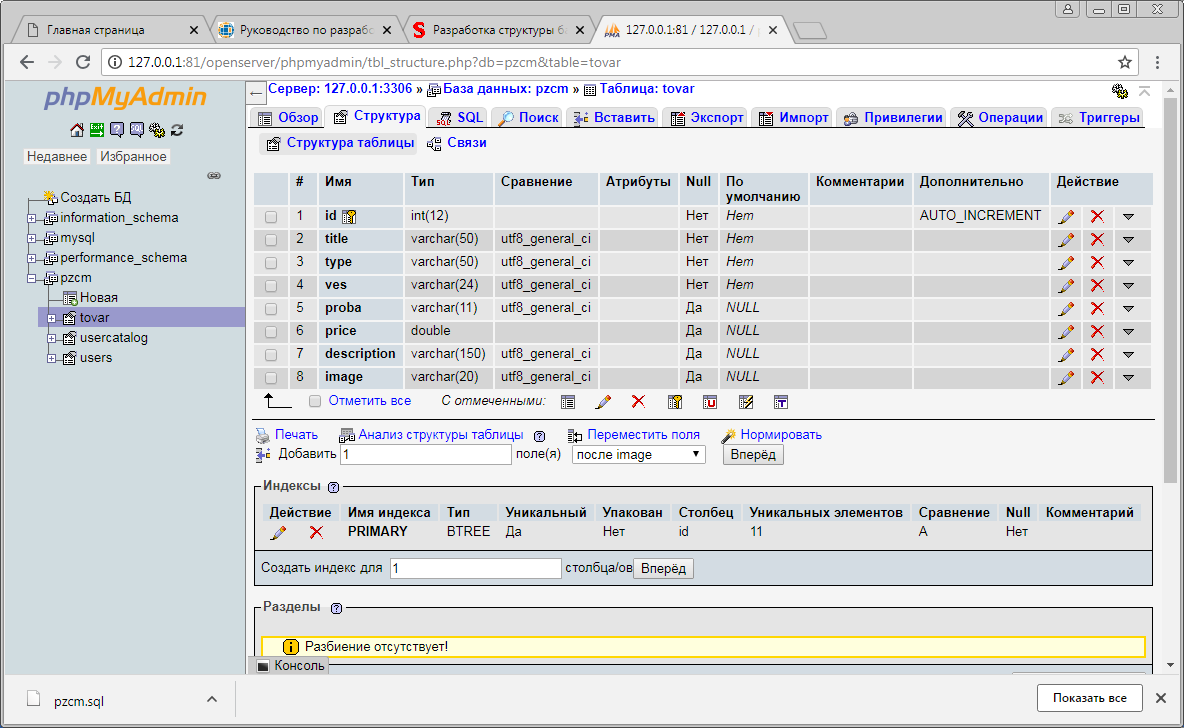
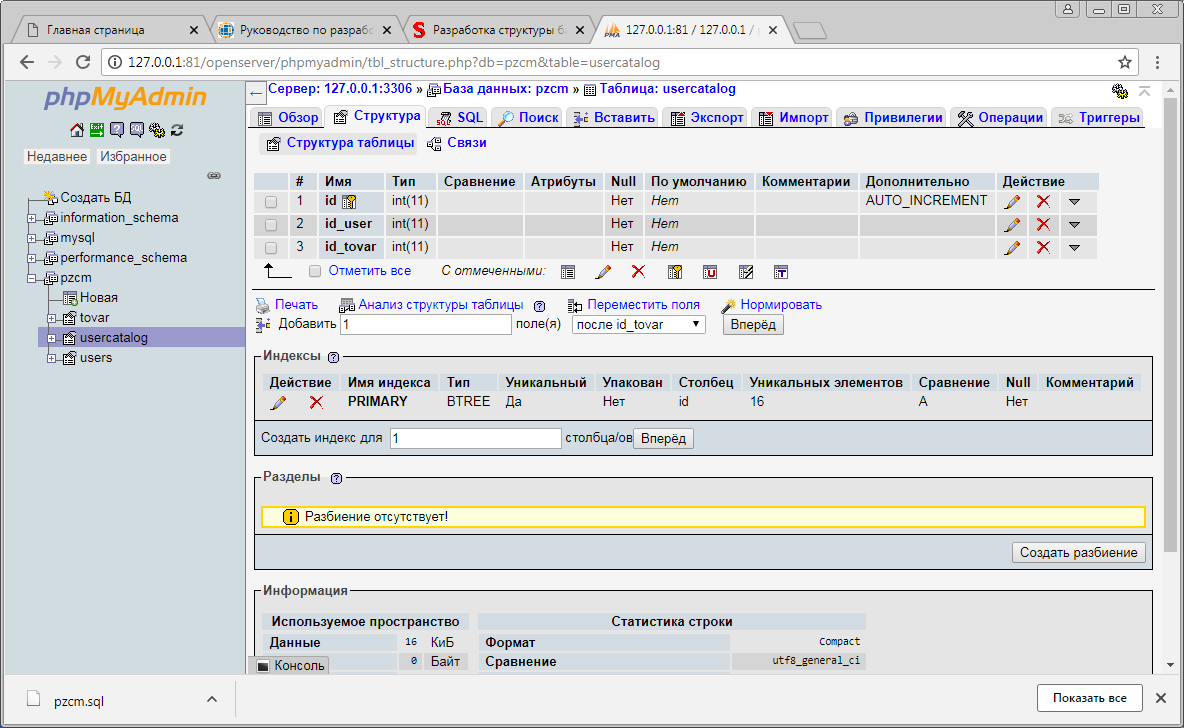


Таблица USERCATALOG (пользовательский каталог) – предназначена для соответствия таблиц товаров и пользователей (таблица 3). Поля USERID и TOVARID являются первичными и внешними ключами таблицы. На основе данной таблицы формируются пользовательские каталоги, которые состоят из записей основного каталога, отобранных тем или иным пользователем.

Таблица 3 – Атрибуты таблицы USERCATALOG (таблица соответствия записей таблицы товаров и таблицы пользователей для формирования пользовательского каталога)



# 5 Описание логической структуры сайта

Один из основных этапов создания сайта – это разработка его структуры. Именно она определяет расположение и взаимосвязь всех элементов ресурса: разделов, подразделов, страниц.

Сайт с продуманной структурой удобен для конечного пользователя, потому что в нем легко ориентироваться. Такой сайт легко развивать, так как в хорошо построенную структуру удобно добавлять новые элементы.

Общая структура разрабатываемого сайта-каталога представлена на рисунке 4.



Рисунок 4 – Общая структура разрабатываемого сайта-каталога

*Описание файлов:*

* index.php – главная страница сайта;
* menu.inc.php – файл, реализующий меню сайта;
* top.inc.php – файл, реализующий шапку сайта;
* bottom.inc.php – файл, реализующий подвал сайта;
* catalog.php – файл основного каталога, отображающий все записи основного каталога и позволяющий удалять выбранные записи администратору, а пользователю – добавлять их в свой каталог;
* add.php – файл, реализующий добавление записи в основной каталог;
* edit.php – файл, реализующий редактирование записи основного каталога;
* item.php – файл, реализующий просмотр записи как основного каталога, так и пользовательского;
* auth.php – файл авторизации, блокирующий доступ к страницам неавторизованным пользователям;
* style.css – файл каскадной таблицы стилей;
* lib.inc.php – файл библиотеки пользовательских функций;
* registration.php – файл регистрации, позволяющий пользователям регистрироваться на сайте и получать доступ к авторизации;
* base\_reg.php – файл регистрации базы данных, хранящий настройки подключения к базе данных.
* profile.php – файл, реализующий редактирование профиля;
* usercatalog.php – файл пользовательского каталога, отображающий все записи авторизованного пользователя в его каталоге и позволяющий удалять выбранные записи из своего пользовательского каталога;
* userlist.php – файл, реализующий просмотр всех зарегистрированных пользователей.

Сайт-каталог, разрабатываемый в рамках курсовой работы, представляет собой совокупность двух подсистем («Администратор» и «Пользователь»), обладающих различными функциональными возможностями. Рассмотрим каждую из подсистем отдельно.

## **5.1 Подсистема «Пользователь»**

Подсистема «Пользователь» может выполнять следующие действия:

1. регистрация на сайте;
2. авторизация;
3. просмотр и редактирование собственного профиля;
4. просмотр основного каталога;
5. сортировка записей основного каталога;
6. поиск записей в основном каталоге;
7. просмотр списка записей пользователя;
8. добавление записей в список пользователя;
9. удаление записей из списка пользователя.

## **5.2 Подсистема «Администратор»**

Помимо всех действий, доступных пользователю администратор также имеет право на изменение данных в основном тематическом информационном сайте учебного заведения, то есть может добавлять, редактировать и удалять записи.

# 6 Экспериментальная часть

## **6.1 Функционирование подсистемы «Пользователь»**

Для получения доступа к информации сайта необходимо пройти авторизацию, то есть ввести свои учетные данные (логин и пароль). Форма авторизации приведена на рисунке 5.

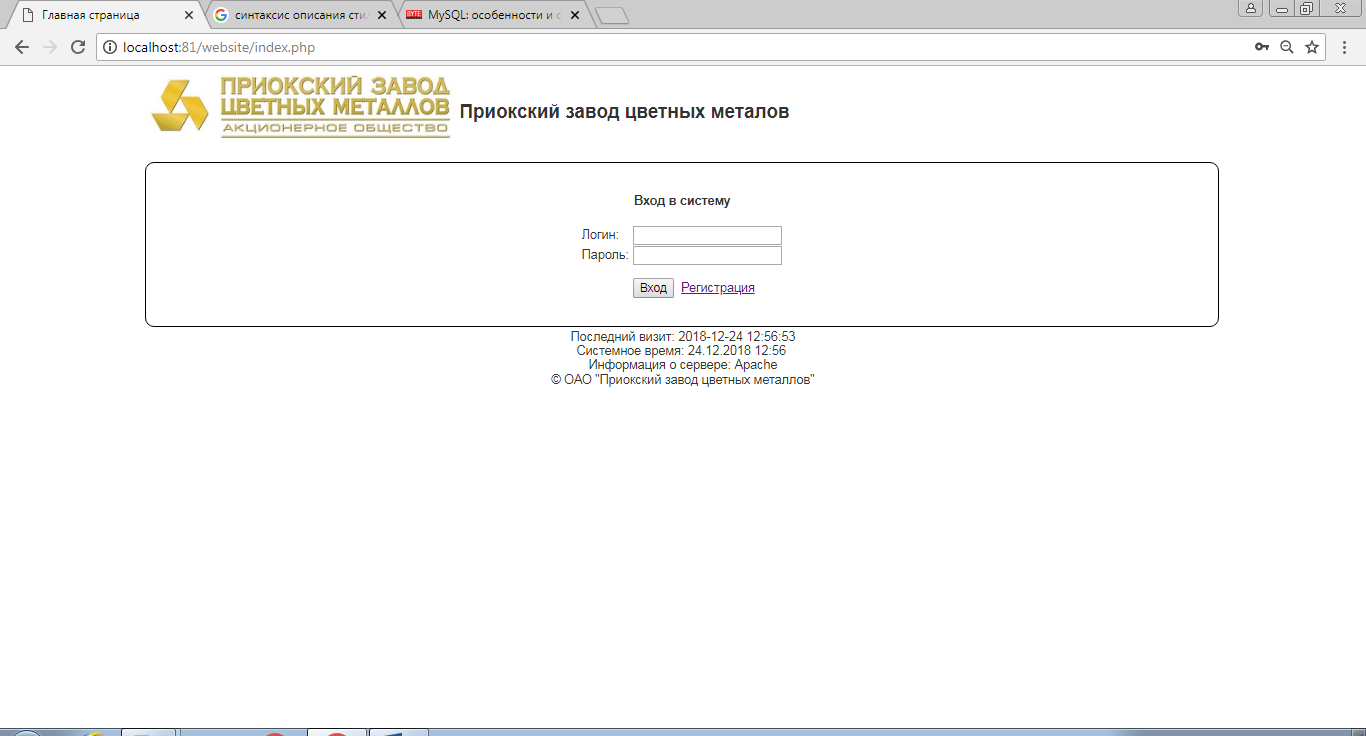


Рисунок 5 – Форма авторизации

В случае если пользователь не зарегистрирован на сайте, он должен предварительно пройти процедуру регистрации. Как выглядит форма регистрации показано на рисунке 6.

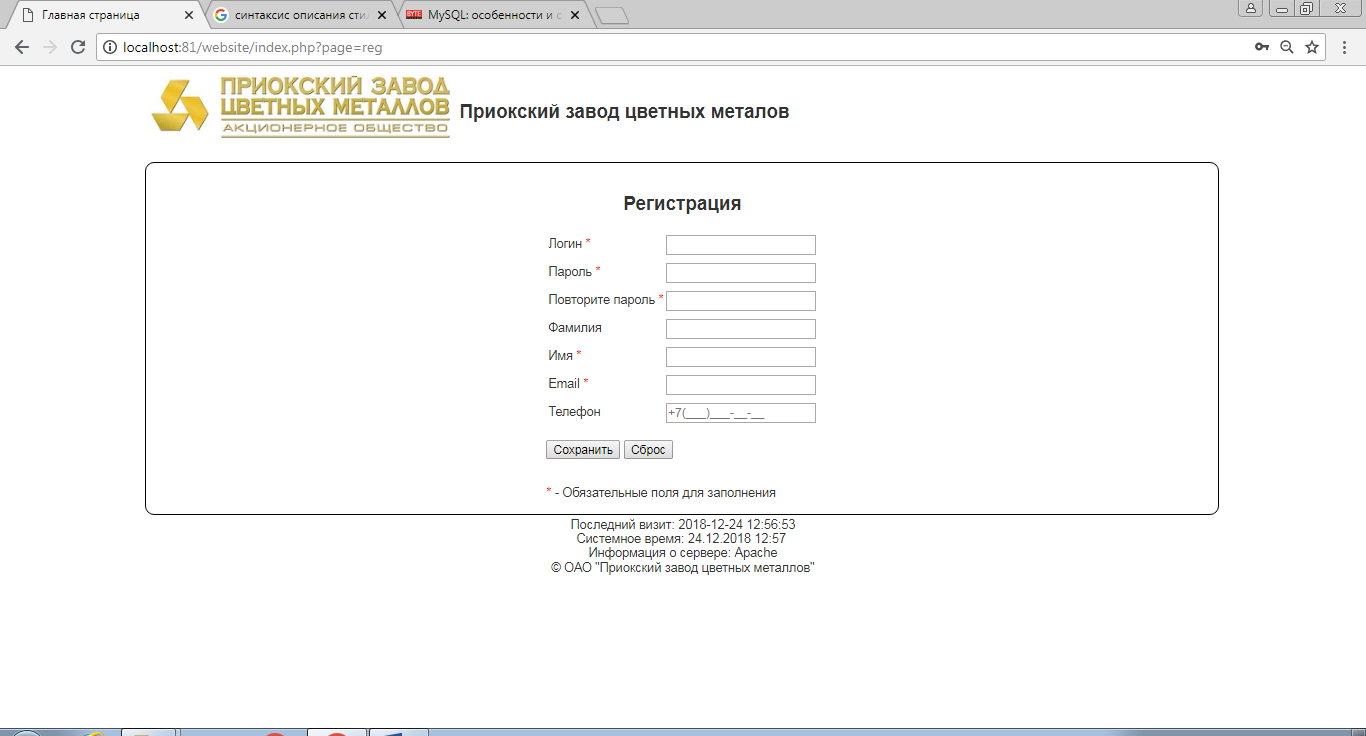


Рисунок 6 – Форма регистрации пользователя

В случае успешной авторизации осуществляется переход на главную страницу сайта (рисунок 7).

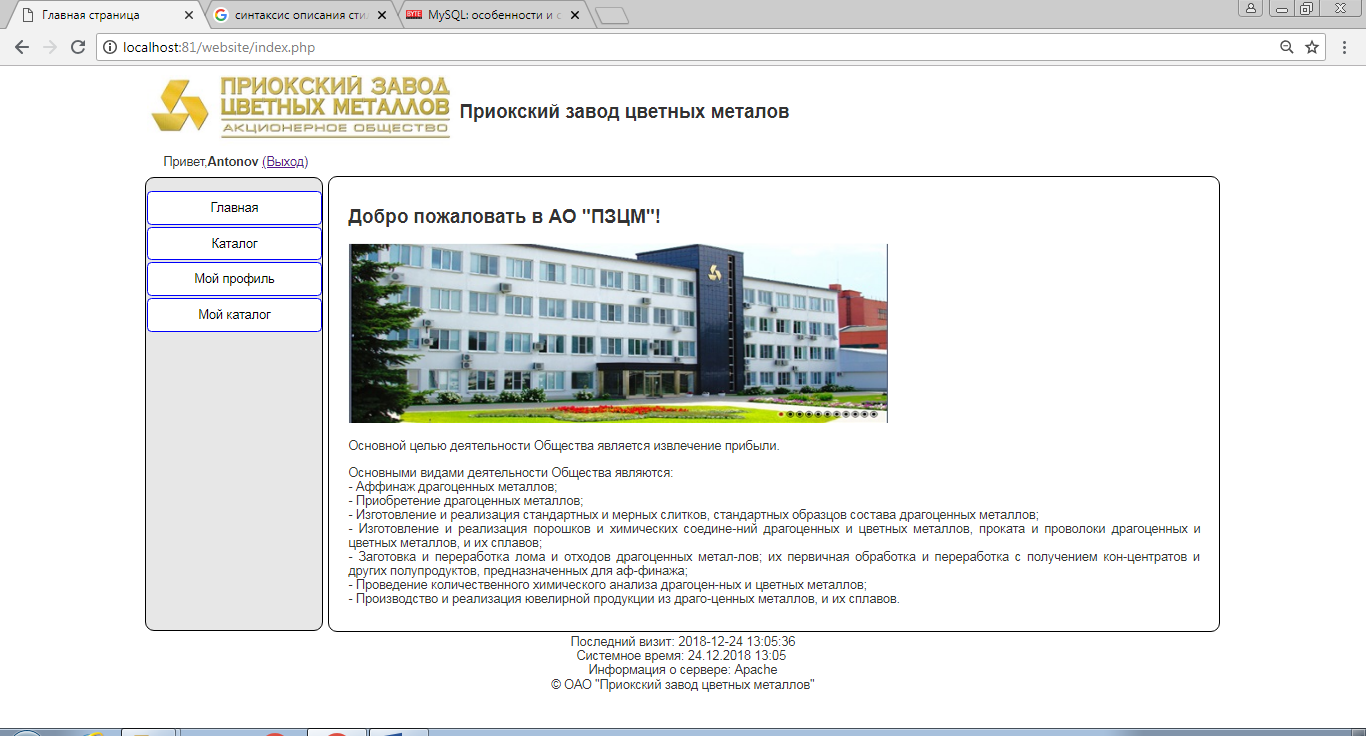


Рисунок 7 – Главная страница сайта (пользователь)

В противном случае выводится сообщение об ошибке и повторный запрос учетных данных (рисунок 8).

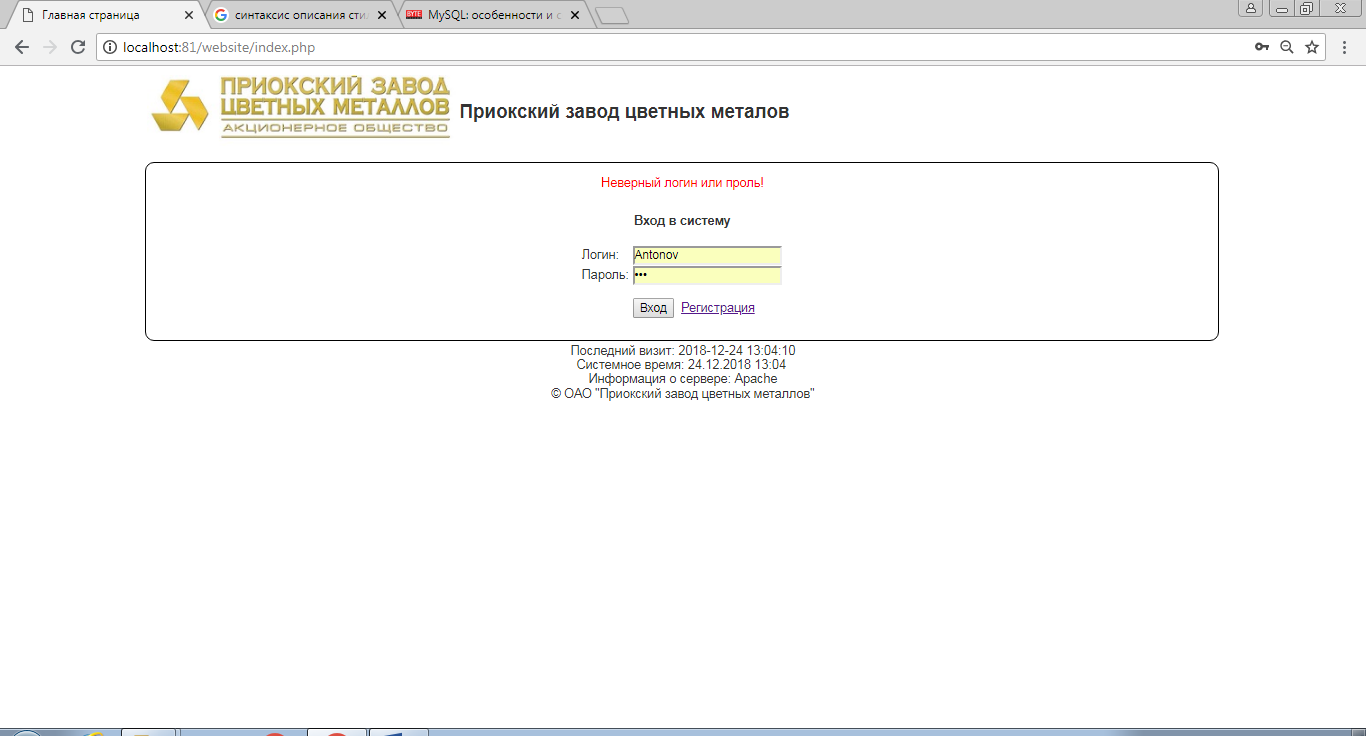


Рисунок 8 – Сообщение об ошибке авторизации

В левой части сайта расположено меню. Для пользователя меню выглядит следующим образом:

1. Главная страница – отображает общую информацию по тематике сайта. Является стартовой страницей сайта и отображается сразу после успешной авторизации.
2. Каталог – список продаваемой продукции Приокским заводом цветных металлов. Содержит список изделий из драгоценных металлов и их характеристики.
3. Мой профиль – отображает регистрационную информацию авторизованного пользователя.
4. Мой каталог – отображает записи каталога, которые авторизованный пользователь добавил из общего каталога в свой пользовательский (рисунок 9).

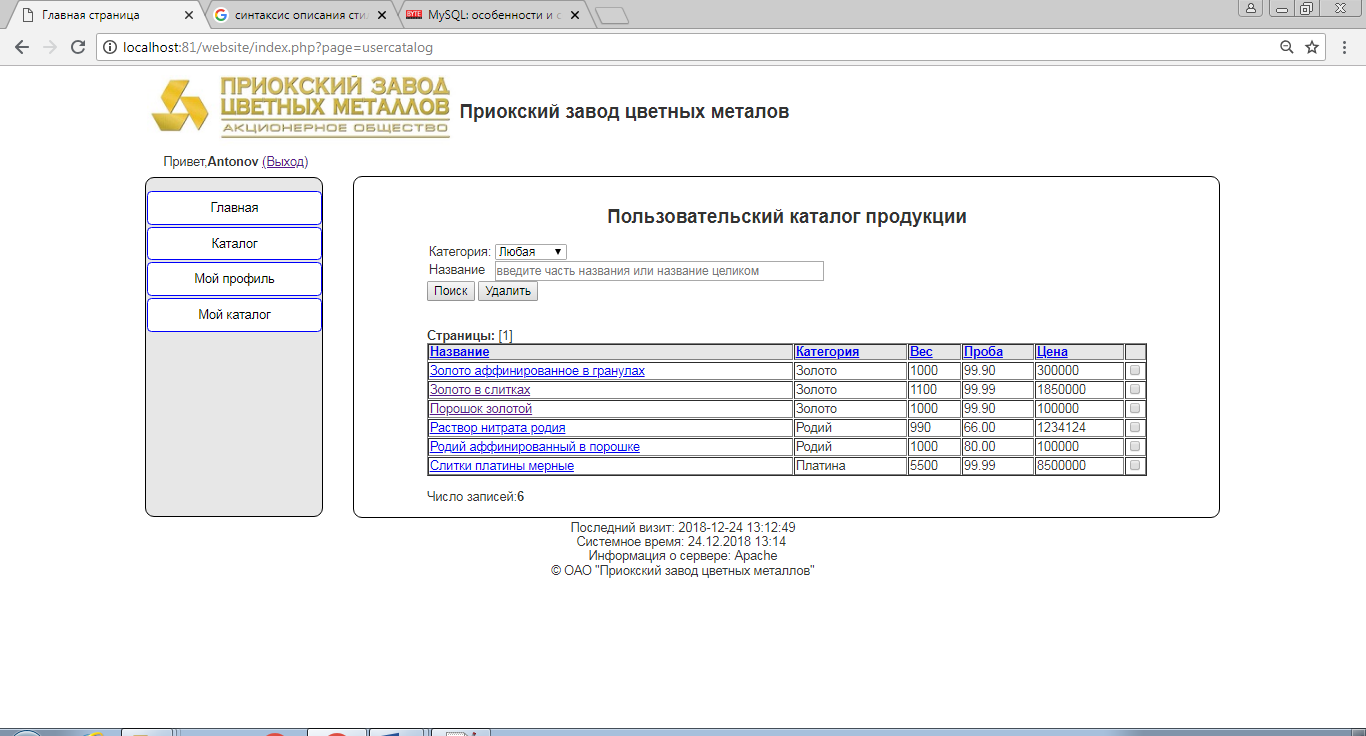


Рисунок 9 – Пользовательский каталог сайта

При переходе по ссылке «Каталог» открывается список продукции, предлагаемой Приокским заводом цветных металлов (рисунок 10).



Рисунок 10 – Общий каталог сайта (вход осуществлен от пользователя)

Для пользователя доступны следующие действия:

* сортировка каталога – сортировка выполняется при переходе по соответствующей ссылке в заголовке таблицы, в данном примере записи отсортированы по пробе изделия;
* поиск по сайту – поиск возможен по категории и названию изделия;
* просмотр записи каталога – просмотр осуществляется нажатием на определённое наименование изделия, которое представляет собой ссылку;
* добавление товара в свой пользовательский каталог.

Для осуществления поиска необходимо из раскрывающегося списка выбрать категорию изделия или ввести хотя бы частичное наименование изделия, либо заполнить оба поля для усиления поиска. Выполним поиск, который должен отобразить все изделя категории «Платина» и содержащие в наименовании «слитки». Для этого в поле категория выберем «Платина», а в поле наименование «слитки», и нажмём кнопку поиск. Результат поиска показан на рисунке 11.

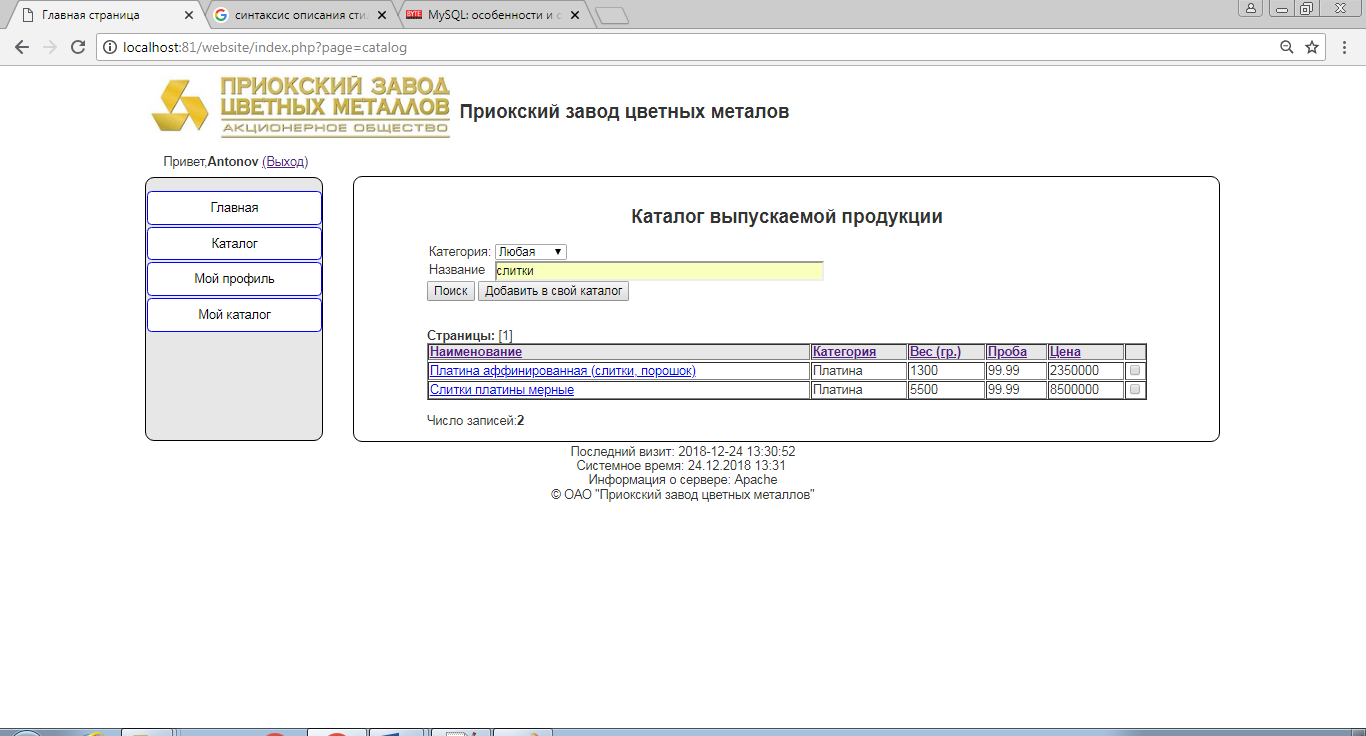


Рисунок 11 – Результат поиска всех изделий из категории «Платина» и с составляющей наименование «слитки»

Каждое наименование изделий в каталоге представляет собой ссылку, перейдя по которой можно получить более подробные характеристики интересующего изделия, а также просмотреть её изображение (рисунок 12).

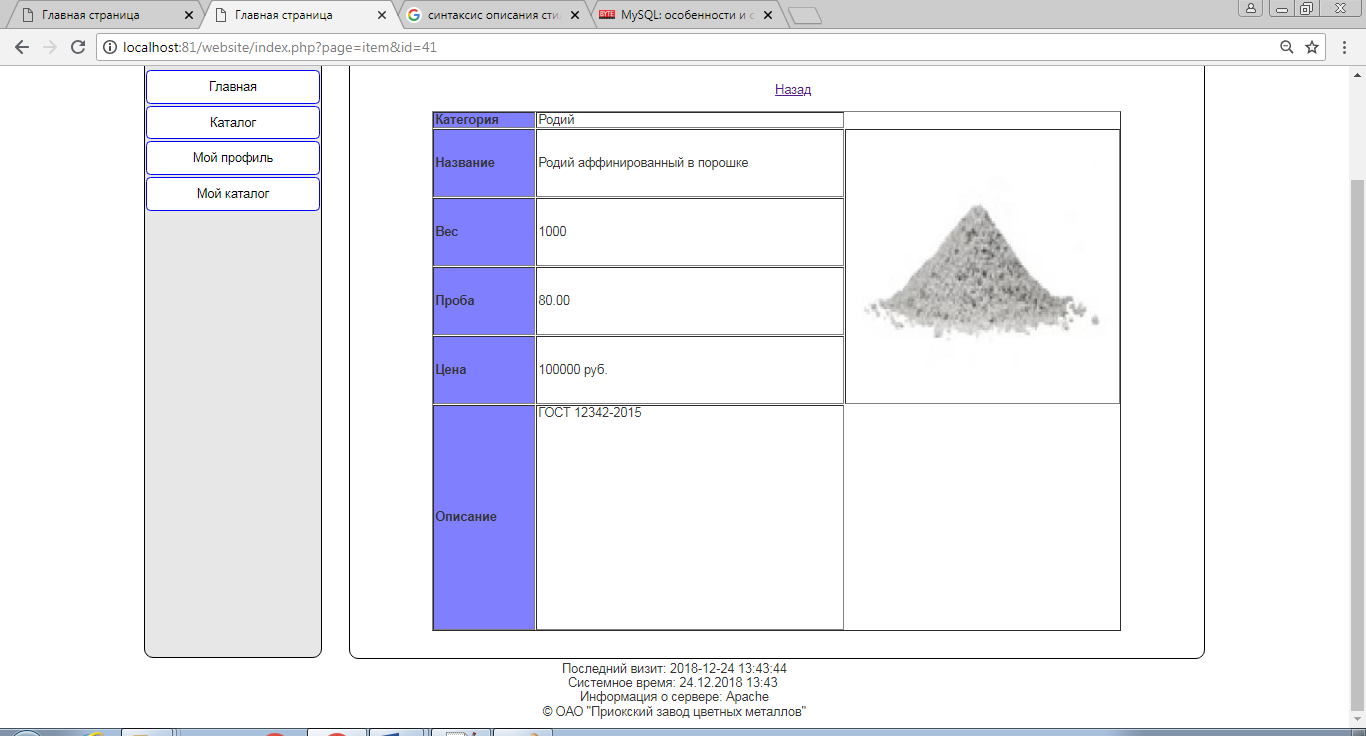


Рисунок 12 – Просмотр записи каталога

Также пользователь может добавлять записи в собственный каталог из основного каталога сайта. Для этого на форме основного каталога нужно выделить необходимые записи и нажать кнопку добавить в свой каталог. Для просмотра своего пользовательского каталога необходимо перейти по ссылке в меню «Мой каталог» (рисунок 13).

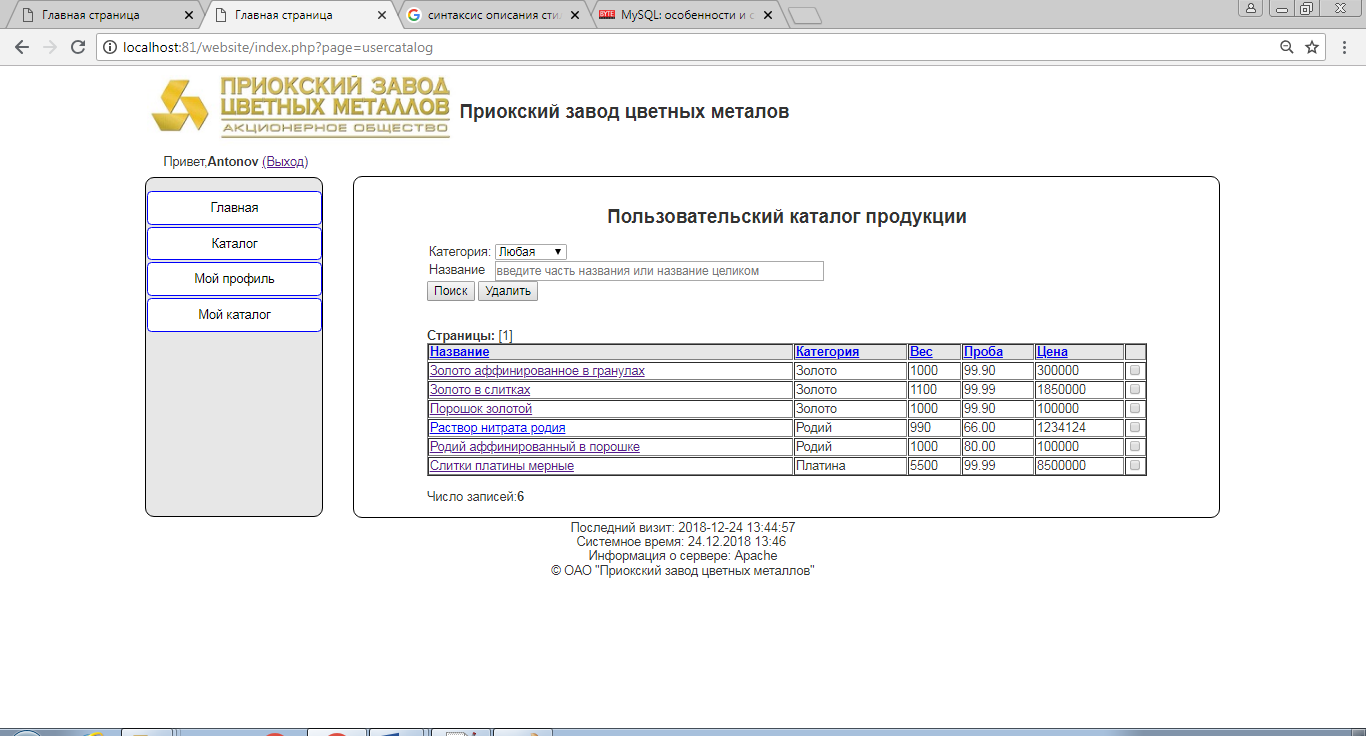


Рисунок 13 – Просмотр пользовательского каталога товаров

В собственном каталоге пользователь может осуществлять следующие действия:

* сортировка пользовательского каталога – сортировка выполняется при переходе по соответствующей ссылке в заголовке таблицы;
* поиск по собственному каталогу – поиск возможен по категории и полному/частичному наименованию искомой продукции;
* просмотр записи каталога – просмотр осуществляется нажатием на определённое название модели, которое представляет собой ссылку;
* удаление товара из своего пользовательского каталога.

Действия по редактированию, добавлению и удалению записей основного каталога пользователь производить не может. Однако, пользователь может редактировать информацию о собственном профиле. Для этого необходимо перейти по ссылке «Мой профиль» в основном меню сайта. В результате открывается форма с регистрационными данными пользователя, введенными при регистрации. Пользователь может их изменить и сохранить изменения в БД, нажав на кнопку «Сохранить» (рисунок 14).

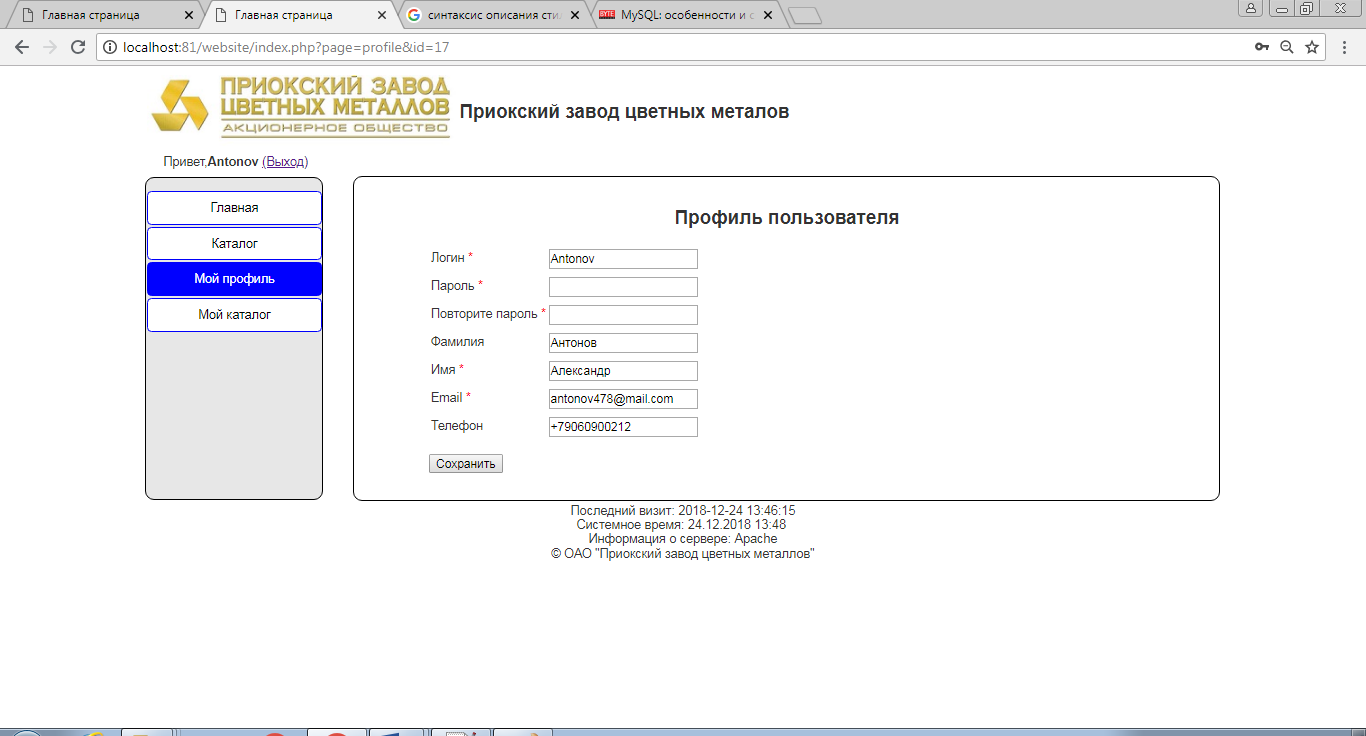


Рисунок 13 – Редактирование профиля пользователя

## **6.2 Функционирование подсистемы «Администратор»**

Для входа на сайт с правами администратора необходимо на форме ввести администраторский логин и пароль, который хранится не в базе данных, а в конфигурационном файле. Для данного сайта логин для администратора – sysdba, а пароль – masterkey.

Меню администратора несколько отличается от меню пользователя: в место ссылки «Мой профиль» для администратора доступен пункт «Список пользователей» (рисунок 15).

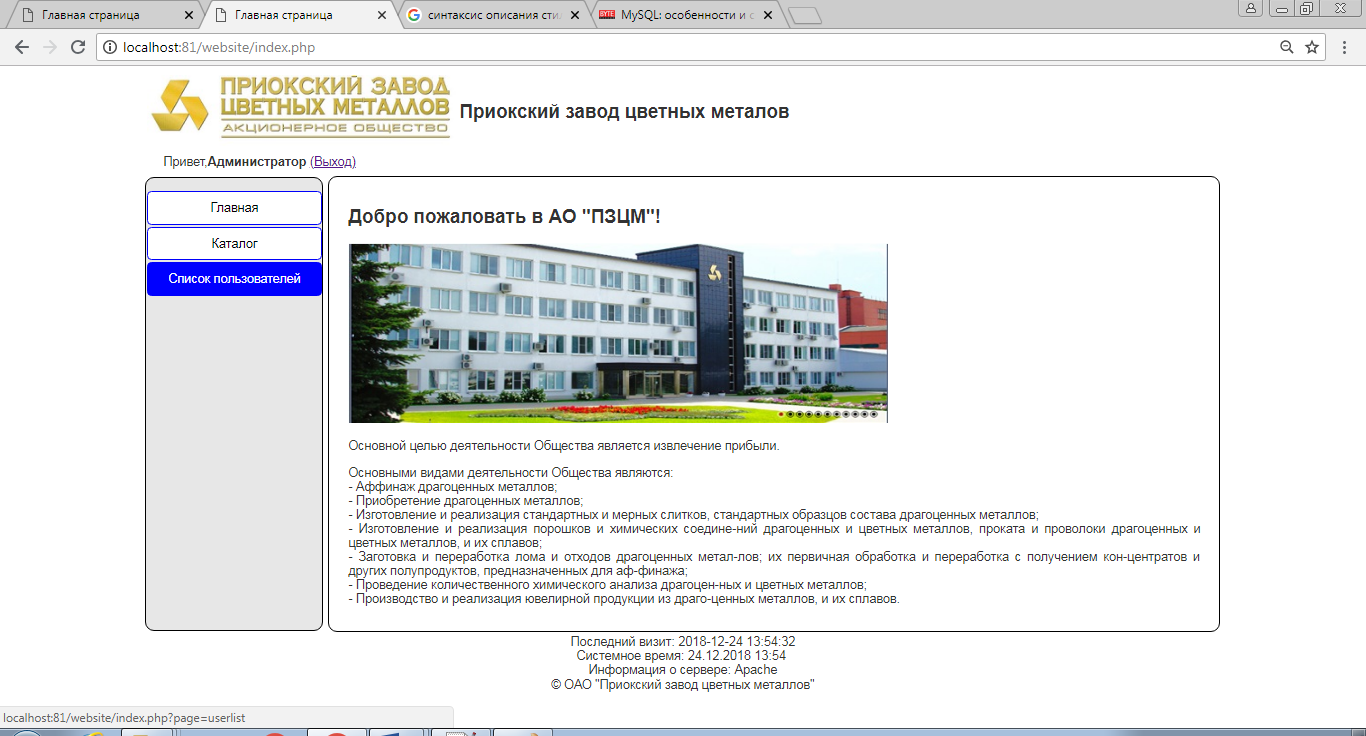


Рисунок 15 – Главная страница сайта (администратор)

При переходе по данной ссылке открывается список всех пользователей, зарегистрированных на данном сайте (рисунок 16).



Рисунок 16 – Список зарегистрированных пользователей

Администратор имеет право изменять регистрационные данные любого пользователя. Для этого нужно нажать на интересующий логин, который представляет собой ссылку. В результате откроется форма с регистрационными данными выбранного пользователя. Администратор может их изменить и сохранить изменения в БД, нажав на кнопку «Сохранить» (рисунок 17).

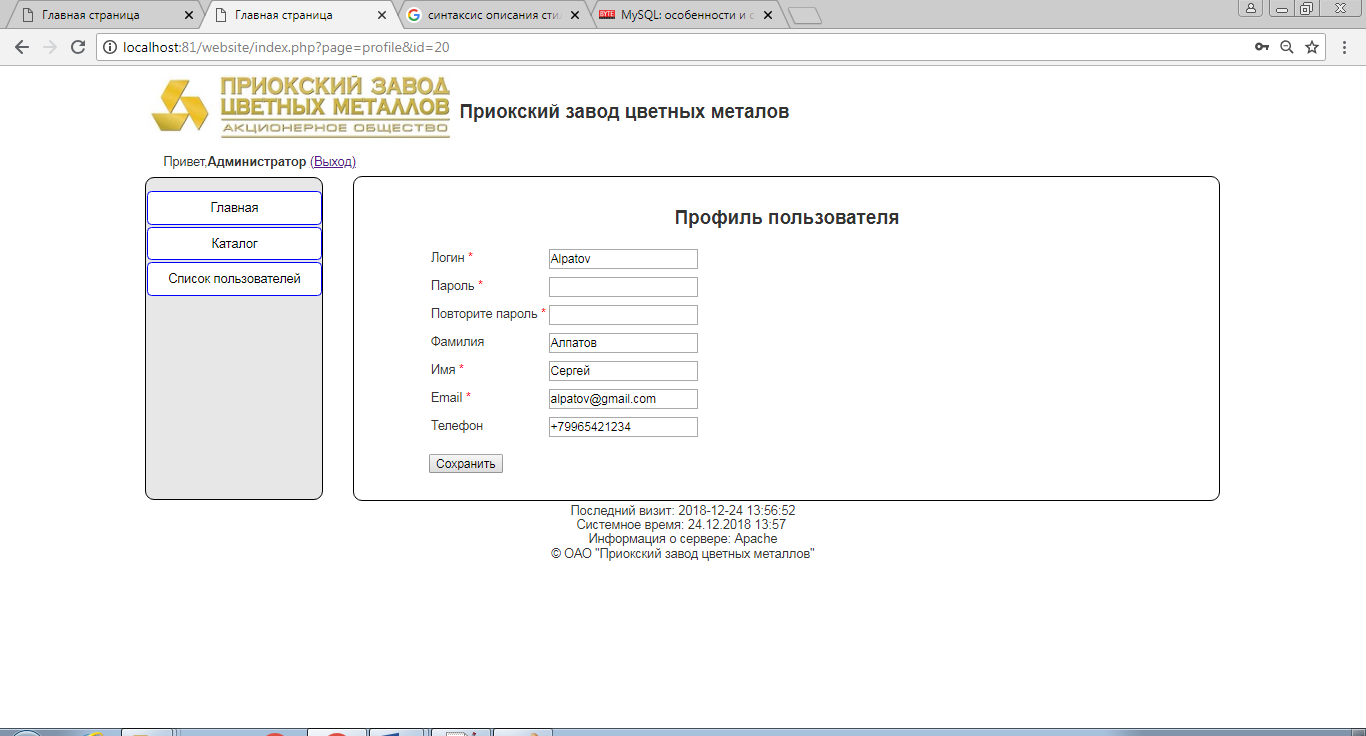


Рисунок 17 – Редактирование профиля пользователя

Так же для администратора доступны действия по редактированию основного каталога (добавление/удаление/редактирование записи). На форме просмотра основного каталога сайта, становятся активными кнопки «Добавить» и «Удалить» (рисунок 18).



Рисунок 18 – Основной каталог сайта (вход осуществлен от администратора)

При нажатии на кнопку добавить, открывается форма добавления записи (рисунок 19).

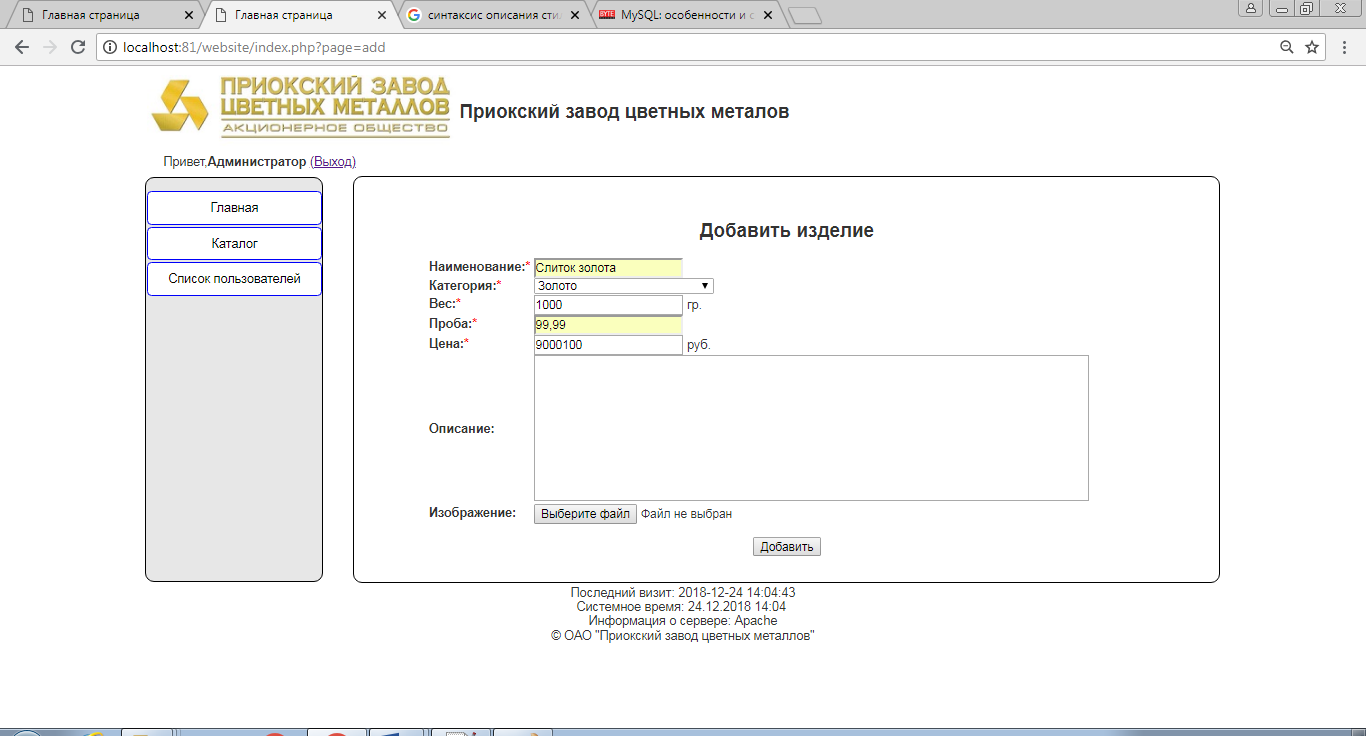


Рисунок 19 – Форма добавления изделия

После ввода всех обязательных для заполнения полей, помеченных красной звёздочкой, при нажатии на кнопку добавить, введённая информация будет добавлена в БД (рисунок 20).

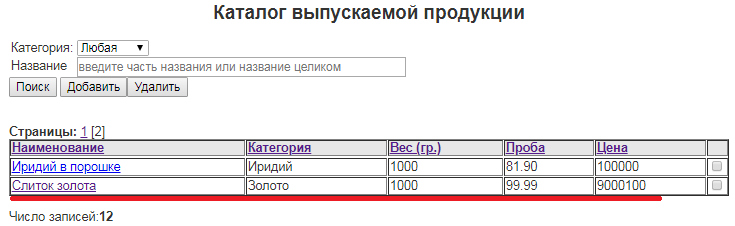


Рисунок 20 – Добавленная запись

Для удаления записей из основного каталога, необходимо отметить ненужные записи галочкой и нажать кнопку удалить (рисунок 21).

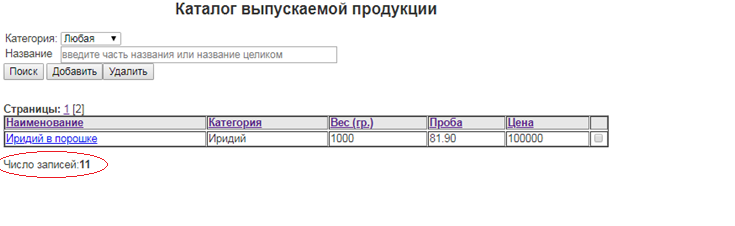


Рисунок 21 – Удаление записи

На форме просмотра записи, становится активной ссылка редактировать, перейдя по которой открывается форма редактирования записей основного каталога (рисунок 22).

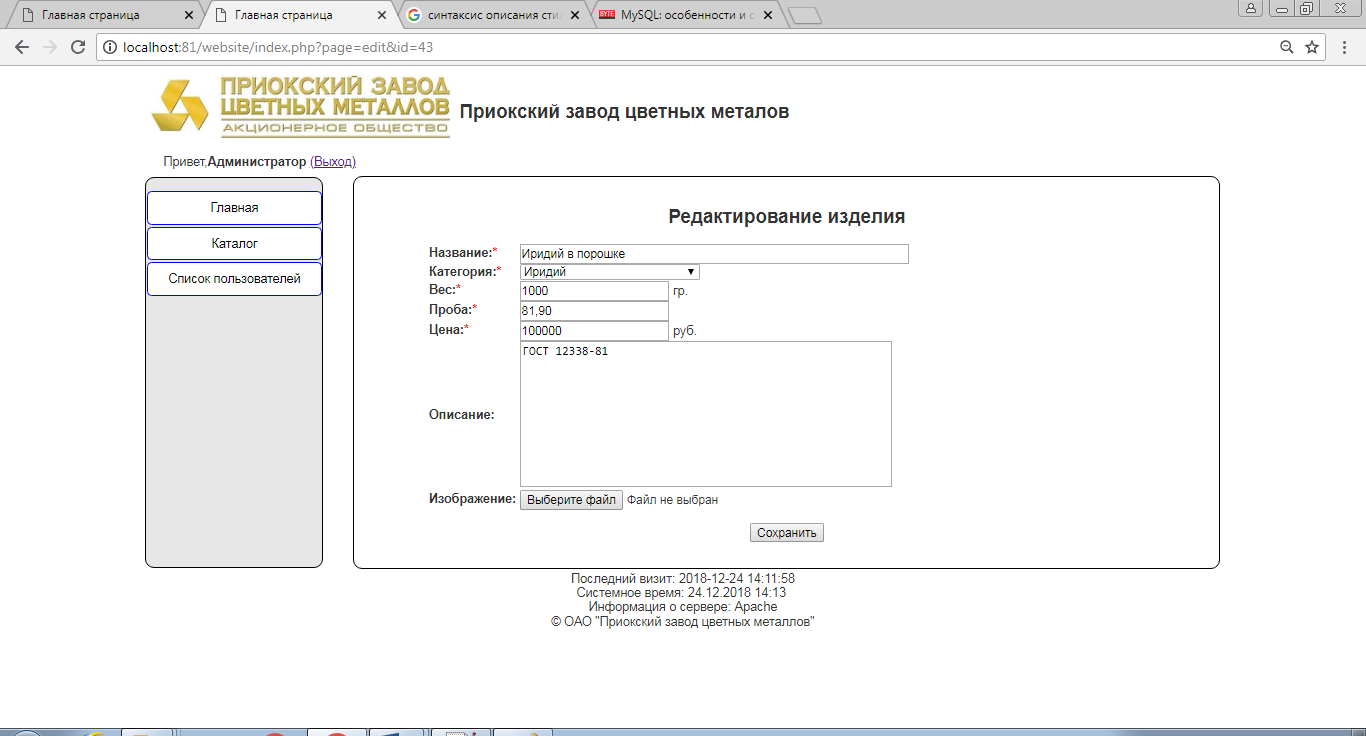


Рисунок 22 – Форма редактирования записи основного каталога

# Заключение

В курсовой работе разработан сайт-каталог для целей реализации продукции из драгоценных металлов, содержащий информацию о изделиях, выпускаемых Приокским заводом цветных металлов.

При выполнении курсовой работы были приобретены следующие навыки:

* установка и настройка веб-сервера Apache и модуля PHP;
* разработка валидных веб-страниц на языке гипертекстовой разметки XHTML и в их верстка с помощью каскадных таблиц стилей CSS;
* разработка одностраничного интерфейса, работа с заголовками, формами, куками и сессиями;
* разработка SQL-запросов, и их выполнение средствами PHP.

Результатом выполнения курсовой работы стал сайт-каталог для целей реализации продукции из драгоценных металлов, содержащий информацию о изделиях, выпускаемых Приокским заводом цветных металлов, реализующий работу в двух режимах – режиме пользователя и администратора. Пользователь и администратор обладают различными правами и возможностями в соответствии с заданием на разработку.

# Список использованных источников

1. Рэдлайн. Web сайты. <https://lred.ru/web> Дата обращения: 20.12.2018.
2. Роль веб-сайта в научно-образовательной деятельности вуза <https://cyberleninka.ru/article/n/rol-veb-sayta-v-nauchno-obrazovatelnoy-deyatelnosti-vuza> Дата обращения: 20.12.2018.
3. Сайт о правильных сайтах. Расширяемый язык разметки гипертекста XHTML <http://cssblok.ru/tech/xhtml.html> Дата обращения: 20.12.2018.
4. Сайт о правильных сайтах. Каскадные листы стилей CSS http://cssblok.ru/tech/css.html Дата обращения: 20.12.2018.
5. CSS. https://ru.wikipedia.org/wiki/CSS. Дата обращения: 21.12.2018.
6. Apache HTTP Server: <https://ru.wikipedia.org/wiki/Apache_HTTP> \_Server. Дата обращения: 19.12.2018.
7. Маркин А.В., Шкарин С.С. Основы web-программирования на PHP. – М.: Диалог-МИФИ, 2012. – 252 с.
8. Мерсер Д.У., Мерсер Д., Кент А., Новицки С., Скуайер Д., Чой В.К. PHP 5 для начинающих.: Пер. с англ. – М.: ООО И.Д. «Вильямс», 2006. – 848 с.
9. MySQL: [Электронный ресурс]. URL:  https://ru.wikipedia.org/wiki/MySQL. (Дата обращения: 18.12.2018).

# Приложение А

**Текст модулей сайта и инструкция по развертыванию**

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

ФАКУЛЬТЕТ АВТОМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ В УПРАВЛЕНИИ

КАФЕДРА АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ

КУРСОВАЯ РАБОТА

**«Разработка сайта-каталога»**

Текст модулей сайта и инструкция по развертыванию

КР–09.04.02–735М–02–18

Листов 29

Рязань 2018

**Аннотация**

В приложении приведен текст модулей разработанного сайта и инструкция по развертыванию сайта.

Программная реализация осуществлена на языке программирования PHP 5.5 с использованием СУБД – MySQL 5.6 и веб-сервера – Apache 2.4.

**Содержание**

[А.1 Текст модулей сайта 42](#_Toc532661539)

[А.1.1 Файл INDEX.PHP 42](#_Toc532661540)

[А.1.2 Файл TOP.INC.PHP 42](#_Toc532661541)

[А.1.3 Файл BOTTOM.INC.PHP 45](#_Toc532661542)

[А.1.4 Файл LIB.INC.PHP 45](#_Toc532661543)

[А.1.5 Файл BASE\_REGISTRATION.PHP 46](#_Toc532661544)

[А.1.6 Файл AUTH.PHP 49](#_Toc532661545)

[А.1.7 Файл REGISTRATION.PHP 49](#_Toc532661546)

[А.1.8 Файл MENU.INC.PHP 52](#_Toc532661547)

[А.1.9 Файл PROFILE.PHP 52](#_Toc532661548)

[А.1.10 Файл CATALOG.PHP 53](#_Toc532661549)

[А.1.11 Файл ADD.PHP 54](#_Toc532661550)

[А.1.12 Файл EDIT.PHP 56](#_Toc532661551)

[А.1.13 Файл ITEM.PHP 58](#_Toc532661552)

[А.1.14 Файл USERLIST.PHP 60](#_Toc532661553)

[А.1.15 Файл USERCATALOG.PHP 60](#_Toc532661554)

[А.1.16 Файл STYLE.CSS 62](#_Toc532661555)

[А.2 Инструкция по развертыванию 66](#_Toc532661556)

# А.1 Текст модулей сайта

## 

## **А.1.1 Файл INDEX.PHP**

<?php

session\_start();

ob\_start();

ini\_set('display\_errors',1);

//error\_reporting(E\_ALL);

date\_default\_timezone\_set('Asia/Muscat');

header("Content-Type: text/html; charset=utf-8");

header("Cache-control: no-store");

include "base\_reg.php";

include "lib.inc.php";

if (isset($\_COOKIE['dateVisit']))

$dateVisit = $\_COOKIE['dateVisit'];

setcookie('dateVisit',date('Y-m-d H:i:s'),time()+0xFFFFFFF);

if (empty($\_GET['page']))

$page = "";

else

$page = $\_GET['page'];

?>

<!DOCTYPE html>

<html lang="en">

<head>

<title>Главная страница</title>

<link rel="stylesheet" href="stylea.css" type="text/css" media="screen" />

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

</head>

<body>

<table class="header">

<tr>

<?php include "top.inc.php" ?>

</tr>

<tr>

<?php if (!empty($\_SESSION['user\_login'])) include "menu.inc.php" ?>

<td>

<table class="content">

<tr>

<td class="content\_td">

<?php

if ($page=="reg")

require 'registration.php';

else

require 'auth.php';

if (isset($\_SESSION['user\_login']))

{

if (empty($page)) {

 echo '<div>  <h2>Добро пожаловать в АО "ПЗЦМ"!</h2>  <p><img src="images/12.jpg" alt="Фото" style="width:600px; height:200px;"/></p>

 <p>Основной целью деятельности Общества является извлечение прибыли.</p>

 <p>Основными видами деятельности Общества являются:

<br> - Аффинаж драгоценных металлов;

<br> - Приобретение драгоценных металлов;

<br> - Изготовление и реализация стандартных и мерных слитков, стандартных образцов состава драгоценных металлов;

<br> - Изготовление и реализация порошков и химических соедине-ний драгоценных и цветных металлов, проката и проволоки драгоценных и цветных металлов, и их сплавов;

<br> - Заготовка и переработка лома и отходов драгоценных метал-лов; их первичная обработка и переработка с получением кон-центратов и других полупродуктов, предназначенных для аф-финажа;

<br> - Проведение количественного химического анализа драгоцен-ных и цветных металлов;

<br> - Производство и реализация ювелирной продукции из драго-ценных металлов, и их сплавов.</span></p>

 </div>';

 }

 else

 {  switch($page)

  {

   case 'catalog': include  'catalog.php';break;

 case 'add': include 'add.php'; break;

 case 'item': include 'item.php'; break;

 case 'edit': include 'edit.php'; break;

  case 'userlist': include 'userlist.php'; break;

 case 'profile': include 'profile.php'; break;

 case 'usercatalog': include 'usercatalog.php'; break;

 }

 }

}

 ?>

 </td>

 </tr>

 </table>

 </td>

  </tr>

<tr>

 <?php include "bottom.inc.php" ?>

</tr>

</table>

</body>

</html>

<?php

 ob\_end\_flush();

?>

## **А.1.2 Файл TOP.INC.PHP**

<td colspan="2" style="height:15%">

<!-- Верхняя часть сайта -->

<table class="top">

<tr>

<td>

<table style="text-align:left">

<tr>

<td >

<a href="index.php"><img src="images/logo.jpg" alt="Логотип" /></a>

</td>

<td >

<h2> Приокский завод цветных металов </h2>

</td>

</tr>

</table>

</td>

</tr>

<tr>

<td class="top\_left">

<?php

if (!empty($\_SESSION['user\_login']))

echo "Привет,<b>".$\_SESSION['user\_login']."</b> <a href='index.php?logout=true'>(Выход)</a>";

?>

</td>

</tr>

</table>

</td>

## **А.1.3 Файл BOTTOM.INC.PHP**

<td colspan="2" style="height:5%">

<!-- Нижняя часть сайта -->

<table class="footer">

<tr>

<td>

Последний визит: <?=$dateVisit?><br>

Системное время: <?=date("d.m.Y H:i")?>

<br/> Информация о сервере: <?=$\_SERVER['SERVER\_SOFTWARE']?>

</td>

</tr>

<tr>

<td>&copy; ОАО "Приокский завод цветных металлов"</td>

</tr>

</table>

</td>

## **А.1.4 Файл LIB.INC.PHP**

<?php

function getMenu($menu, $vertical=true)

{

$style = "";

if(!$vertical)

{

$style = "display:inline";

}

echo '<ul style="list-style-type:none">';

foreach ($menu as $link=>$href)

{

echo "<li style='$style'><a href=\"$href\">", $link, '</a></li>';

}

echo '</ul>';

}

function clearData($data)

{

return trim(strip\_tags($data));

}

function loadImage($p){ // функция принимает параметр, который получает значение либо add(новое изображение) либо edit(замена изображения)

$a = array("mess"=>"","src"=>""); // возвращаемый массив из двух элементов $a['mess'] - сообщение об ошибке если есть и $a['src'] - путь к изображению

if ($\_FILES['uploadfile']['type'] != 'image/jpeg')

$a['mess']= '<font color="red" align="center" >Не верный тип изображения!</font>';

else

{

if ($\_FILES['uploadfile']['size'] > 100000)

$a['mess'] = '<font color="red" align="center" >Размер изображения слишком большой! (макс.=100кб.)</font>';

else

{

$Image = imagecreatefromjpeg($\_FILES['uploadfile']['tmp\_name']); // создаем изображение

$Size = getimagesize($\_FILES['uploadfile']['tmp\_name']); // получаем размер изображения

$Tmp = imagecreatetruecolor(300, 300);

imagecopyresampled($Tmp, $Image, 0, 0, 0, 0, 300, 300, $Size[0], $Size[1]); // изменяем размер на 300 на 300 $Size[0] и $Size[1] текущие размеры выбранного изображения

if ($p=="add")

$Download = 'images/catalog/'.count($\_SESSION['catalog']);

else

$Download = $\_SESSION['catalog'][$\_GET['id']]['image'];

if (empty($Download)) $Download = 'images/catalog/'.$\_GET['id'];

imagejpeg($Tmp, $Download.'.jpg'); // сохраняем изображение на сервере в каталоге $Download

imagedestroy($Image);

imagedestroy($Tmp);

$a['src']=$Download;

}

}

return $a;

}

$connect = false;

function connectDB(){ // функция подключения к БД

global $connect;

$host = "127.0.0.1";

$user = "root";

$password = "";

$database = "pzcm";

$connect = mysqli\_connect($host,$user,$password,$database) or die("Не удалось подключиться к БД"); // подключение к базе данных

/\* изменение набора символов на utf8 \*/

mysqli\_set\_charset($connect,"utf8");

}

function getOne($query){ // функция получения одной записи из БД

global $connect;

connectDB();

$result\_set = mysqli\_query($connect,$query) or die("Ошибка " . mysqli\_error($connect));;

closeDB();

return $result\_set->fetch\_assoc();

}

function getAll($query){ // функция получения нескольких записей из БД

global $connect;

connectDB();

$result\_set = mysqli\_query($connect,$query) or die("Ошибка " . mysqli\_error($connect));;

closeDB();

return resultSetArray($result\_set);

}

function executeQuery($query){ // функция выполнения запросов, которые не возвращают данные (INSERT,UODATE,DELETE и т.д)

global $connect;

connectDB();

$result\_set = mysqli\_query($connect,$query) or die("Ошибка " . mysqli\_error($connect));;

closeDB();

}

function resultSetArray($result\_set){ // функция преобразования полученных данных из БД в ассоциативный массив

$array =array();

while (($row = $result\_set->fetch\_assoc()) !=false)

$array[] = $row;

return $array;

}

function closeDB() { // закрытие соединения с БД

global $connect;

mysqli\_close($connect);

}

function addUser($login, $password, $surname, $name, $email, $phone){ // функция добавления пользователя в таблицу users

global $connect;

connectDB();

$query ="SELECT id FROM users where login = '$login' or email = '$email'"; // проверяем нет ли в таблице пользователя с таким логином или емаилом

$result = getOne($query);

if(!empty($result))

{

echo "<font color=red>Пользователь с таким логином или email уже существует!</font>";

}

else

{

$query ="INSERT INTO users (login, password, surname, name, email, phone) VALUES ('$login', '$password', '$surname', '$name', '$email','$phone')";

executeQuery($query);

header("Location: index.php?login=".$login);

}

}

function editUser($id,$login, $password\_main, $password\_confirm, $surname, $name, $email, $phone){ // функция добавления пользователя в таблицу users

$fl = 1;

$password = "";

if (!empty($password\_main) || !empty($password\_confirm))

{

if ($password\_main == $password\_confirm)

{

$password = md5(md5($password\_main.'salt')); // хэшируем пароль

}

else

{

echo '<font color=red>Пароли не совпадают!</font>';

$fl = 0;

}

}

if ($fl == 1)

{

global $connect;

connectDB();

$query ="SELECT id FROM users where id <> $id and (login = '$login' or email = '$email')"; // проверяем нет ли в таблице пользователя с таким логином или емаилом

$result = getOne($query);

if(!empty($result))

{

echo "<font color=red>Пользователь с таким логином или email уже существует!</font>";

}

else

{

if ($password != "")

$query ="UPDATE users set login = '$login', password = '$password', surname = '$surname', name = '$name', email = '$email', phone = '$phone' WHERE id = $id";

else

$query ="UPDATE users set login = '$login', surname = '$surname', name = '$name', email = '$email', phone = '$phone' WHERE id = $id";

executeQuery($query);

echo "<font color=green>Данные сохранены!</font>";

}

}

}

function login($login,$password){

global $connect;

connectDB();

}

function makePager($kol, $all\_count, $page) // простая пагенация

{

// помещаем номер страницы из массива GET в переменую $nom

$nom = isset($\_GET["nom"]) ? (int) $\_GET["nom"] : 1;

// количество записей на страницу

$on\_page = $kol;

// (номер страницы - 1) \* записей на страницу

$shift = ($nom - 1) \* $on\_page;

// получаем количество записей

$count = $all\_count;

$pages = ceil($count / $on\_page);

for ($i = 1; $i <= $pages; $i++) {

// если текущая старница

if($i == $nom){

echo " [$i] ";

} else {

echo "<a href='index.php?page=".$page."&nom=$i'>$i</a> ";

}

}

return $shift;

}

?>

## **А.1.5 Файл BASE\_REGISTRATION.PHP**

<?php

$host="localhost";

$user="root";

$password="";

$database="pzcm";

$admin = array("login" => "sysdba", "password" => "masterkey");

?>

## **А.1.6 Файл AUTH.PHP**

<?php

if (isset($\_POST['login']) && isset($\_POST['password']))

{

if (!empty($\_POST['login']) && !empty($\_POST['password']))

{

$login = clearData($\_POST['login']);

$password = clearData($\_POST['password']);

if ($login == $admin['login'] && $password == $admin['password'])

{

$\_SESSION['user\_login'] = "Администратор";

$\_SESSION['ip'] = $\_SERVER['REMOTE\_ADDR'];

header("Location: http://".$\_SERVER['HTTP\_HOST'].$\_SERVER['REQUEST\_URI']);

exit;

}

else

{

$password = md5(md5($password.'salt'));

$res = getOne("SELECT id from users where login='$login' and password='$password'");

if (!empty($res)){

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

$\_SESSION['ip'] = $\_SERVER['REMOTE\_ADDR'];

$\_SESSION['user\_id'] = $res['id'];

header("Location: http://".$\_SERVER['HTTP\_HOST'].$\_SERVER['REQUEST\_URI']);

exit;

}

else

echo "<font color=red>Неверный логин или проль!</font>";

}

}

else

echo "<font color=red>Заполните все поля!</font>";

}

if (isset($\_GET['logout']))

{

session\_destroy();

header("Location: index.php");

exit;

}

if (!isset($\_SESSION['user\_login']) and $\_SESSION['ip'] != $\_SERVER['REMOTE\_ADDR'])

{

?>

<table class="content">

<tr>

<td>

<h4>Вход в систему</h4>

<form method="POST">

<center>

<table>

<tr>

<td>

Логин:

</td>

<td>

<input type="text" name="login" value="<?php echo $\_GET['login']; ?>" required><br>

</td>

</tr>

<tr>

<td>

Пароль:

</td>

<td>

<input type="password" name="password" required><br>

</td>

</tr>

<tr>

<td></td>

<td>

<p><input type="submit" value="Вход">&nbsp&nbsp<a href='index.php?page=reg'>Регистрация</a></p>

</td>

</tr>

</table>

</center>

</form>

</td>

</tr>

</table>

<?php

}

?>

## **А.1.7 Файл REGISTRATION.PHP**

<?php

if ($\_SERVER['REQUEST\_METHOD'] == 'POST')

{

if (!empty($\_POST['password\_main']) && !empty($\_POST['password\_confirm']) && !empty($\_POST['login']) && !empty($\_POST['email']) && !empty($\_POST['name']))

{

$password\_main = clearData($\_POST['password\_main']);

$password\_confirm = clearData($\_POST['password\_confirm']);

if ($password\_main == $password\_confirm)

{

$password = md5(md5($password\_main.'salt')); // хэшируем пароль

$login = clearData($\_POST['login']);

$surname = clearData($\_POST['surname']);

$name = clearData($\_POST['name']);

$email = clearData($\_POST['email']);

$phone = clearData($\_POST['phone']);

addUser($login, $password, $surname, $name, $email, $phone);

}

else echo '<font color=red>Пароли не совпадают!</font>';

}

else echo '<font color=red>Заполните все обязательные поля!</font>';

}

?>

<center>

<table>

<tr>

<td>

<h2 align="center">Регистрация</h2>

<form method="POST">

<table>

<tr style="height:30px;">

<td>Логин <font color="red">\*</font></td>

<td><input type="text" name="login" required></td>

</tr>

<tr style="height:30px;">

<td>Пароль <font color="red">\*</font></td>

<td><input type="password" name="password\_main" required></td>

</tr>

<tr style="height:30px;">

<td>Повторите пароль <font color="red">\*</font></td>

<td><input type="password" name="password\_confirm" required></td>

</tr>

</tr>

<tr style="height:30px;">

<td>Фамилия</td>

<td><input type="text" name="surname"></td>

</tr>

<tr style="height:30px;">

<td>Имя <font color="red">\*</font></td>

<td><input type="text" name="name" required></td>

</tr>

<tr style="height:30px;">

<td>Email <font color="red">\*</font></td>

<td><input type="text" name="email" required></td>

</tr>

<tr style="height:30px;">

<td>Телефон</td>

<td><input type="text" name="phone" placeholder="+7(\_\_\_)\_\_\_-\_\_-\_\_"></td>

</tr>

</table>

<p>

<input type="submit" value="Сохранить">

<input type="reset" value="Сброс">

</p>

<br>

<font color="red">\*</font> - Обязательные поля для заполнения

</form>

</td>

</tr>

</table>

</center>

## **А.1.8 Файл MENU.INC.PHP**

<?php

if ($\_SESSION['user\_login'] == "Администратор")

$menu = array(

"Главная"=>"index.php",

"Каталог"=>"index.php?page=catalog",

"Список пользователей"=>"index.php?page=userlist");

else

$menu = array(

"Главная"=>"index.php",

"Каталог"=>"index.php?page=catalog",

"Мой профиль"=>"index.php?page=profile&id=".$\_SESSION['user\_id'],

"Мой каталог"=>"index.php?page=usercatalog");

?>

<td style="height:100%">

<table class="menu">

<tr>

<td>

<?php

getMenu($menu);

?>

</td>

</tr>

</table>

</td>

## **А.1.9 Файл PROFILE.PHP**

<?php

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

if ($\_SERVER['REQUEST\_METHOD'] == 'POST')

{

if (!empty($\_POST['login']) && !empty($\_POST['email']) && !empty($\_POST['name']))

{

$login = clearData($\_POST['login']);

$surname = clearData($\_POST['surname']);

$name = clearData($\_POST['name']);

$email = clearData($\_POST['email']);

$phone = clearData($\_POST['phone']);

$password\_main = clearData($\_POST['password\_main']);

$password\_confirm = clearData($\_POST['password\_confirm']);

editUser($id,$login, $password\_main, $password\_confirm, $surname, $name, $email, $phone);

}

else echo '<font color=red>Заполните все обязательные поля!</font>';

}

$res = getOne("SELECT \* from users where id=$id");

?>

<div class="data">

<table class="table">

<tr>

<td>

<h2 align="center">Профиль пользователя</h2>

<form action="index.php?page=profile&id=<?php echo $id; ?>" method="POST">

<table>

<tr style="height:30px;">

<td>Логин <font color="red">\*</font></td>

<td><input type="text" name="login" value="<?php echo $res['login']; ?>" required></td>

</tr>

<tr style="height:30px;">

<td>Пароль <font color="red">\*</font></td>

<td><input type="password" name="password\_main" ></td>

</tr>

<tr style="height:30px;">

<td>Повторите пароль <font color="red">\*</font></td>

<td><input type="password" name="password\_confirm" ></td>

</tr>

</tr>

<tr style="height:30px;">

<td>Фамилия</td>

<td><input type="text" name="surname" value="<?php echo $res['surname']; ?>"></td>

</tr>

<tr style="height:30px;">

<td>Имя <font color="red">\*</font></td>

<td><input type="text" name="name" value="<?php echo $res['name']; ?>" required></td>

</tr>

<tr style="height:30px;">

<td>Email <font color="red">\*</font></td>

<td><input type="text" name="email" value="<?php echo $res['email']; ?>" required></td>

</tr>

<tr style="height:30px;">

<td>Телефон</td>

<td><input type="text" name="phone" placeholder="+7(\_\_\_)\_\_\_-\_\_-\_\_" value="<?php echo $res['phone']; ?>"></td>

</tr>

</table>

<p>

<input type="submit" value="Сохранить">

</p>

</form>

</td>

</tr>

</table>

</div>

## **А.1.10 Файл CATALOG.PHP**

<?php

if (isset($\_POST['add'])) header("Location: index.php?page=add");

if (isset($\_POST['del'])){

if (!empty($\_POST['delId'])){

foreach($\_POST['delId'] as $val)

{

executeQuery("DELETE FROM tovar WHERE id = '$val'"); // удаляем записи из таблицы tovar

 @unlink('images/catalog/'.$val.'.jpg');

 executeQuery("DELETE FROM usercatalog WHERE id\_tovar = '$val'"); // удаляем записи из таблицы tovar

}

}

 else echo "<font color='red'>Сначала отметьте записи, которые необходимо удалить!</font>";

}

if (isset($\_POST['addtome'])){

if (!empty($\_POST['delId'])){

$id\_user = $\_SESSION['user\_id'];

foreach($\_POST['delId'] as $val)

{

executeQuery("INSERT INTO usercatalog (id\_user,id\_tovar) value ($id\_user, $val)"); // удаляем записи из  таблицы tovar

}

}

  Else echo "<font color='red'>Сначала отметьте записи, которые хотите добавить в свой каталог!</font >";

}

if (isset($\_SESSION['sort'])) // поле сортировки сортировка

{

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

if (isset($\_GET['sort']))

{

switch ($\_GET['sort']){

case "title":$\_SESSION['sort'] = "title"; break;

case "type":$\_SESSION['sort'] = "type"; break;

case "price":$\_SESSION['sort'] = "price"; break;

case "ves":$\_SESSION['sort'] = "ves"; break;

case "proba":$\_SESSION['sort'] = "proba"; break;

}

}

}

else

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

$where ="";

if (isset($\_POST['search'])) // поиск

{

$type = clearData($\_POST['type']);

$title = clearData($\_POST['title']);

if ($type == "Любая")

{

if (!empty($title))

$where = "WHERE lower(title) like lower('%".$title."%') ";

}

if (empty($title))

{

if ($type != "Любая")

$where = "WHERE type = '".$type."' ";

}

if (!empty($title) && $type!="Любая")

$where = "WHERE type = '".$type."' and lower(title) like lower('%".$title."%') ";

}

    SC";

$rows = getAll($query);

?>

<div class="data">

<center><h2>Каталог выпускаемой продукции</h2></center>

<form action="index.php?page=catalog" method="POST">

<table>

<tr>

<td>Категория:</td>

<td>

<select size="1" name="type">

<option value="Любая">Любая</option>

<option value="Золото">Золото</option>

<option value="Серебро">Серебро</option>

<option value="Платина">Платина</option>

<option value="Паладий">Паладий</option>

<option value="Родий">Родий</option>

<option value="Иридий">Иридий</option>

<option value="Рутений">Рутений</option>

</select>

</td>

</tr>

<tr>

<td>Название</td>

<td><input type="text" name="title" placeholder="введите часть названия или название целиком" size="50"></td>

</tr>

</table>

<input type="submit" name="search" value="Поиск">

<?php

if ($\_SESSION['user\_login'] == "Администратор")

{

echo '<input type="submit" name="add" value="Добавить"><input type="submit" name="del" value="Удалить">';

}

else

?>

<br><br><br>

<b>Страницы:</b>

<?php

$all\_count = count($rows); // получаем общее количество записей

$on\_page = 10; // кол-во записей на странице

$shift = makePager($on\_page, $all\_count, "catalog"); // формируем пагенацию

$rows = getAll($query." LIMIT $shift, $on\_page");

?>

<table class="table" border="1">

<tr>

<th><a href="index.php?page=catalog&sort=type">Наименование</a></th>

<th><a href="index.php?page=catalog&sort=titel">Категория</a></th>

<th><a href="index.php?page=catalog&sort=ves">Вес (гр.)</a></th>

<th><a href="index.php?page=catalog&sort=proba">Проба</a></th>

<th><a href="index.php?page=catalog&sort=price">Цена</a></th>

<th></th>

<tr>

<?php

if (!empty($rows)){

foreach($rows as $item)

{

echo "<tr>";

echo "<td><a href='index.php?page=item&id=".$item['id']."'>".$item['title']."</a></td><td>".$item['type']."</td><td>".$item['ves']."</td><td>".$item['proba']."</td><td>".$item['price']."</td>";

echo "<td width='10px'><input type='checkbox' name='delId[]' value=".$item['id']."></td>";

echo "</tr>";

}

}

?>

</table>

<br>

Число записей:<b><?php echo $all\_count; ?></b>

</form>

</div>

## **А.1.11 Файл ADD.PHP**

<?php

if ($\_SERVER['REQUEST\_METHOD'] == 'POST')

{

if (!empty($\_POST['title']) && !empty($\_POST['price']))

{

$title = clearData($\_POST['title']);

$type = clearData($\_POST['type']);

$ves = clearData($\_POST['ves']);

$proba = clearData($\_POST['proba']);

$price = clearData($\_POST['price']);

$description = clearData($\_POST['description']);

$cheakID = getOne("SELECT id FROM tovar WHERE title = '$title' ");

if (count($cheakID) == 0)

{

if ($\_FILES['uploadfile']['tmp\_name'])

{

$a = loadImage("add"); // получаем массив , содержащий сообщение в случае ошибки и ссылку на изображение

if (empty($a['mess']))

{

 $image = $a['src'];

 executeQuery("INSERT INTO Tovar (title,type,ves,proba,price,description,image) VALUES ('$title','$type','$ves','$proba','$price','$description','$image')");

 header("Location: index.php?page=catalog");

 exit;

 }

else

{

echo $a['mess'];

}

}

else

{

$image = "";

 executeQuery("INSERT INTO Tovar (title,type,ves,proba,price,description,image) VALUES ('$title','$type','$ves','$proba','$price','$description','$image')");

header("Location: index.php?page=catalog");

exit;

}

}

else echo '<font color="red">Запись с таким наименованием уже существует!</font>';

}

else

echo '<font color="red">Заполните все поля!</font>';

}

?>

<div class="data">

<center><h2>Добавить изделие</h2></center>

<form method='POST' action='index.php?page=add' ENCTYPE='multipart/form-data'>

<table>

<tr>

<th>Наименование:<font color="red">\*</font></th>

<td><input type='text' name='title' required></td>

</tr>

<tr>

<th>Категория:<font color="red">\*</font></th>

<td style="width:100%">

<select size="1" name="type">

<option disabled>Выберите категорию товара</option>

<option value="Золото">Золото</option>

<option value="Серебро">Серебро</option>

<option value="Платина">Платина</option>

<option value="Паладий">Паладий</option>

<option value="Родий">Родий</option>

<option value="Иридий">Иридий</option>

<option value="Рутений">Рутений</option>

</select>

</td>

</tr>

<tr>

<th>Вес:<font color="red">\*</font></th>

<td><input type='number' step="0.1" name='ves' min="0" required>&nbsp;гр.</td>

</tr>

<tr>

<th>Проба:<font color="red">\*</font></th>

<td><input type='number' step="0.01" name='proba' min="0" required>&nbsp;</td>

</tr>

<tr>

<th>Цена:<font color="red">\*</font></th>

<td><input type='number' step="0.01" name='price' min="0" required>&nbsp;руб.</td>

</tr>

<tr>

<th>Описание:</th>

<td><textarea name='description' rows='10' style="resize:none; width:90%"></textarea></td>

</tr>

<tr>

<th>Изображение:</th>

<td><input type='file' name='uploadfile' accept='image/jpeg'></td>

</tr>

</table>

<center><p><input type='submit' value='Добавить' name='add'></p></center>

</form>

</div>

## **А.1.12 Файл EDIT.PHP**

<?php

$id = clearData($\_GET['id']);

$row = getOne("SELECT \* FROM tovar WHERE id = '$id'"); // получаем всю информацию из tovar по данному id

if ($\_SERVER['REQUEST\_METHOD'] == 'POST')

{

if (!empty($\_POST['title']) && !empty($\_POST['price']))

{

$title = clearData($\_POST['title']);

$type = clearData($\_POST['type']);

$ves = clearData($\_POST['ves']);

$proba = clearData($\_POST['proba']);

$price = clearData($\_POST['price']);

$description = clearData($\_POST['description']);

if ($\_FILES['uploadfile']['tmp\_name'])

{

$a = loadImage("edit"); // получаем массив , содержащий сообщение в случае ошибки и ссылку на изображение

if (empty($a['mess']))

{

$image = $a['src'];

executeQuery("UPDATE tovar SET title = '$title', type = '$type', ves = '$ves', proba = '$proba', price = '$price', description = '$description', image ='$image' WHERE id = '$id'");

header("Location: index.php?page=catalog");

exit;

}

else

{

echo $a['mess'];

}

}

else

{

executeQuery("UPDATE tovar SET title = '$title', type = '$type', ves = '$ves', proba = '$proba', price = '$price', description = '$description' WHERE id = '$id'");

header("Location: index.php?page=catalog");

exit;

}

}

else

echo '<font color="red">Заполните все поля!</font>';

}

?>

<div class="data">

<center><h2>Редактирование изделия</h2></center>

<form method='POST' action='index.php?page=edit&id=<?php echo $id; ?>' ENCTYPE='multipart/form-data'>

<table>

<tr>

<th>Название:<font color="red">\*</font></th>

<td><input type='text' name='title' value='<?=$row['title']?>' size="60" required></td>

</tr>

<tr>

<th>Категория:<font color="red">\*</font></th>

<td>

<select size="1" name="type">

<option disabled>Выберите категорию товара</option>

<option value="Золото" <?php if ($row['type'] == "Золото") echo "selected" ?> >Золото</option>

<option value="Серебро" <?php if ($row['type'] == "Серебро") echo "selected" ?>>Серебро</option>

<option value="Платина" <?php if ($row['type'] == "Платина") echo "selected" ?>>Платина</option>

<option value="Паладий" <?php if ($row['type'] == "Паладий") echo "selected" ?>>Паладий</option>

<option value="Родий" <?php if ($row['type'] == "Родий") echo "selected" ?>>Родий</option>

<option value="Иридий" <?php if ($row['type'] == "Иридий") echo "selected" ?>>Иридий</option>

<option value="Рутений" <?php if ($row['type'] == "Рутений") echo "selected" ?>>Рутений</option>

</select>

</td>

</tr>

<tr>

<th>Вес:<font color="red">\*</font></th>

<td><input type='number' step="0.1" name='ves' value='<?=$row['ves']?>' required>&nbsp;гр.</td>

</tr>

<tr>

<th>Проба:<font color="red">\*</font></th>

<td><input type='number' step="0.01" name='proba' value='<?=$row['proba']?>' required>&nbsp;</td>

</tr>

<tr>

<th>Цена:<font color="red">\*</font></th>

<td><input type='number' step="0.01" name='price' value='<?=$row['price']?>' required>&nbsp;руб.</td>

</tr>

<tr>

<th>Описание:</th>

<td><textarea name='description' rows='10' cols='50' style="resize:none;" ><?=$row['description']?></textarea></td>

</tr>

<tr>

<th>Изображение:</th>

<td><input type='file' name='uploadfile' "></td>

</tr>

</table>

<center><p><input type='submit' value='Сохранить'></p></center>

</form>

</div>

## **А.1.13 Файл ITEM.PHP**

<?php

if ($\_SERVER['REQUEST\_METHOD'] == 'GET')

{

$id = clearData($\_GET['id']);

$row = getOne("SELECT \* FROM tovar WHERE id = '$id'"); // получаем всю информацию из tovar по данному id

}

?>

<br/>

<a href='index.php?page=catalog' style='margin-left:40px'>Назад</a>

<?php

if ($\_SESSION['user\_login'] == "Администратор")

echo "<a href='index.php?page=edit&id=".$id."' style='margin-left:20px'>Редактировать</a>";

?>

<br/><br/>

<div class="data">

<table border="1" style="text-align:left;" align="center">

<tr>

<th bgcolor="#8080ff">Категория</th>

<td width="45%"><?= $row['type'] ?></td>

</tr>

<tr>

<th width="15%" bgcolor="#8080ff">Название</th>

<td ><?= $row['title'] ?></td>

<td rowspan="4"><img src='<?php if (empty($row['image'])) echo "images/catalog/no-image.jpg"; else echo $row['image'].'.jpg';?> '></td>

</tr>

<tr>

<th width="15%" bgcolor="#8080ff">Вес</th>

<td ><?= $row['ves'] ?></td>

</tr>

<tr>

<th width="15%" bgcolor="#8080ff">Проба</th>

<td ><?= $row['proba'] ?></td>

</tr>

<tr>

<th bgcolor="#8080ff">Цена</th>

<td><?= $row['price'] ?> руб.</td>

</tr>

<tr height="250">

<th width="15%" bgcolor="#8080ff">Описание</th>

<td valign="top"><?= $row['description'] ?></td>

</tr>

</table>

</div>

<br/>

## **А.1.14 Файл USERLIST.PHP**

<?php

if (isset($\_POST['del'])){

if (!empty($\_POST['delId'])){

foreach($\_POST['delId'] as $val)

{

executeQuery("DELETE FROM users WHERE id = '$val'"); // удаляем записи из таблицы tovar

executeQuery("DELETE FROM usercatalog WHERE id\_user = '$val'"); // удаляем записи из таблицы tovar

}

}

else echo "<font color='red'>Сначала отметьте записи, которые необходимо удалить!</font>";

}

if (isset($\_SESSION['sort'])) // поле сортировки сортировка

{

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

if (isset($\_GET['sort']))

{

switch ($\_GET['sort']){

case "login":$sort = "login"; break;

case "surname":$sort = "surname"; break;

case "name":$sort = "name"; break;

case "email":$sort = "email"; break;

case "phone":$sort = "phone"; break;

}

}

}

else

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

$query = "SELECT \* FROM users ORDER BY ".$\_SESSION['sort']." ASC";

$rows = getAll($query);

?>

<div class="data">

<center><h2>Список зарегестрированных пользователей</h2></center>

<form action="index.php?page=userlist" method="POST">

<input type="submit" name="del" value="Удалить">

<br><br>

<b>Страницы:</b>

<?php

$all\_count = count($rows); // получаем общее количество записей

$on\_page = 7; // кол-во записей на странице

$shift = makePager($on\_page, $all\_count, "userlist"); // формируем пагенацию

$rows = getAll($query." LIMIT $shift, $on\_page");

?>

<table class="table" border="1">

<tr>

<th><a href="index.php?page=userlist&sort=login">Логин</a></th>

<th><a href="index.php?page=userlist&sort=surname">Фамилия</a></th>

<th><a href="index.php?page=userlist&sort=name">Имя</a></th>

<th><a href="index.php?page=userlist&sort=email">Email</a></th>

<th><a href="index.php?page=userlist&sort=phone">Телефон</a></th>

<th></th>

<tr>

<?php

if (!empty($rows)){

foreach($rows as $item)

{

echo "<tr>";

echo "<td><a href='index.php?page=profile&id=".$item['id']."'>".$item['login']."</a></td>";

echo "<td>".$item['surname']."</td><td>".$item['name']."</td>";

echo "<td>".$item['email']."</td><td>".$item['phone']."</td>";

echo "<td width='10px'><input type='checkbox' name='delId[]' value=".$item['id']."></td>";

echo "</tr>";

}

}

?>

</table>

<br>

Число записей:<b><?php echo $all\_count; ?></b>

</form>

</div>

## **А.1.15 Файл USERCATALOG.PHP**

<?php

if (isset($\_POST['del'])){

if (!empty($\_POST['delId'])){

$id\_user = $\_SESSION['user\_id'];

foreach($\_POST['delId'] as $val)

{

executeQuery("DELETE FROM usercatalog WHERE id\_tovar = $val and id\_user = $id\_user");

}

}

else echo "<font color='red'>Сначала отметьте записи, которые необходимо удалить!</font>";

}

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

{

$\_SESSION['sort'] = "title";

if (isset($\_GET['sort']))

{

switch ($\_GET['sort']){

case "title":$\_SESSION['sort'] = "title"; break;

case "type":$\_SESSION['sort'] = "type"; break;

case "ves":$\_SESSION['sort'] = "ves"; break;

case "proba":$\_SESSION['sort'] = "proba"; break;

case "price":$\_SESSION['sort'] = "price"; break;

}

}

}

else

$\_SESSION['sort'] = "title";

$where ="WHERE id\_user = ".$\_SESSION['user\_id']." ";

if (isset($\_POST['search']))

{

$type = clearData($\_POST['type']);

$title = clearData($\_POST['title']);

if ($type == "Любая")

{

if (!empty($title))

$where = $where."and lower(title) like lower('%".$title."%') ";

}

if (empty($title))

{

if ($type != "Любая")

$where = $where."and type = '".$type."' ";

}

if (!empty($title) && $type != "Любая")

$where = $where. "and type = '".$type."' and lower(title) like lower('%".$title."%') ";

}

$query = "SELECT tovar.\* FROM usercatalog LEFT JOIN tovar on tovar.id = usercatalog.id\_tovar ".$where."ORDER BY ".$\_SESSION['sort']." ASC";

$rows = getAll($query);

?>

<div class="data">

<center><h2>Пользовательский каталог продукции</h2></center>

<form action="index.php?page=usercatalog" method="POST">

<table>

<tr>

<td>Категория:</td>

<td>

<select size="1" name="type">

<option value="Любая">Любая</option>

<option value="Золото">Золото</option>

<option value="Серебро">Серебро</option>

<option value="Платина">Платина</option>

<option value="Паладий">Паладий</option>

<option value="Родий">Родий</option>

<option value="Иридий">Иридий</option>

<option value="Рутений">Рутений</option>

</select>

</td>

</tr>

<tr>

<td>Название</td>

<td><input type="text" name="title" placeholder="введите часть названия или название целиком" size="50"></td>

</tr>

</table>

<input type="submit" name="search" value="Поиск">

<input type="submit" name="del" value="Удалить">

<br><br><br>

<b>Страницы:</b>

<?php

$all\_count = count($rows); // получаем общее количество записей

$on\_page = 7; // кол-во записей на странице

$shift = makePager($on\_page, $all\_count, "usercatalog"); // формируем пагенацию

$rows = getAll($query." LIMIT $shift, $on\_page");

?>

<table class="table" border="1">

<tr>

<th><a href="index.php?page=usercatalog&sort=title">Название</a></th>

<th><a href="index.php?page=usercatalog&sort=type">Категория</a></th>

<th><a href="index.php?page=usercatalog&sort=ves">Вес</a></th>

<th><a href="index.php?page=usercatalog&sort=proba">Проба</a></th>

<th><a href="index.php?page=usercatalog&sort=price">Цена</a></th>

<th></th>

<tr>

<?php

if (!empty($rows)){

foreach($rows as $item)

{

echo "<tr>";

echo "<td><a href='index.php?page=item&id=".$item['id']."'>".$item['title']."</a></td><td>".$item['type']."</td><td>".$item['ves']."</td><td>".$item['proba']."</td><td>".$item['price']."</td>";

echo "<td width='10px'><input type='checkbox' name='delId[]' value=".$item['id']."></td>";

echo "</tr>";

}

}

?>

</table>

<br>

Число записей:<b><?php echo $all\_count; ?></b>

</form>

</div>

## **А.1.16 Файл STYLE.CSS**

html, body

{

font:14px Arial, Helvetica, sans-serif;

color:#333;

background:#fff;

-moz-background-size: 100%; /\* Firefox 3.6+ \*/

-webkit-background-size: 100%; /\* Safari 3.1+ и Chrome 4.0+ \*/

-o-background-size: 100%; /\* Opera 9.6+ \*/

background-size: 100%; /\* Современные браузеры \*/

text-align:center;

border:1;

}

.header

{

margin:0 auto;

width:80%;

height:100%;

background:#fff;

border-collapse: separate;

}

.top, .content, .menu, .footer

{

width:100%;

height:100%;

}

.menu

{

width:200px;

}

.top

{

border-spacing:0;

}

.top\_left, .top\_right

{

border:1px solid #fff;

padding:5px 20px;

background:#fff;

}

.top\_left

{

border-radius:8px 0 0 8px;

border-right-width:0;

text-align:left;

background:#fff;

}

.top\_right

{

border-radius:0 8px 8px 0;

border-left-width:0;

text-align:right;

background:#fff;

}

.footer

{

text-align:center;

font-size:14px;

background:#fff;

}

.menu td, .content\_td

{

border:1px solid #000;

border-radius:10px;

vertical-align:top;

padding-top:15px;

padding-bottom:15px;

position:relative;

}

.menu td

{

background:#e7e7e7;

text-align:center;

}

.menu ul

{

list-style-type:none;

margin:0px;

padding-left:0px;

}

.menu a

{

text-decoration:none;

color:black;

background:#FFF;

display:block;

padding:10px;

border-radius:5px;

border: 1px solid #0000fa

}

.menu a:hover

{

background:blue;

color:white;

}

.menu li

{

margin-bottom:2px;

}

.content\_td div

{

display:block;

text-align:justify;

padding-left:20px;

padding-right:20px;

width:800;

}

.table

{

width:100%;

}

.table th

{

background:#e7e7e7;

}

.data

{

width:800px;

margin:auto;

}

# А.2 Инструкция по развертыванию

В курсовой работе в качестве веб-интерфейса для администрирования БД используется phpMyAdmin. Локальный веб сервер – Open Server Panel.

Для корректной работы и просмотра сайта рекомендуется:

1. Скачать Open Server Panel с дополнительным инструментом для администрирования БД phpMyAdmin (в basic-дистрибутиве Open Server Panel уже имеется).
2. Запустить Open Server Panel.
3. Далее, необходимо импортировать файл pzcm.sql в базу данных. Для этого нажимаем правой кнопкой мыши на иконку Open Server Panel в области уведомлений и выбираем phpMyAdmin (рисунок 1)

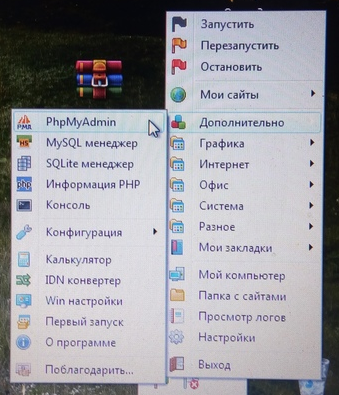


Рисунок 1 – Запуск phpMyAdmin

1. После открытия phpMyAdmin появляется веб-интерфейс входа в программу (рисунок 2). Введем в графе «Пользователь» root, а поле «Пароль» оставим пустым. Язык выберем по желанию.

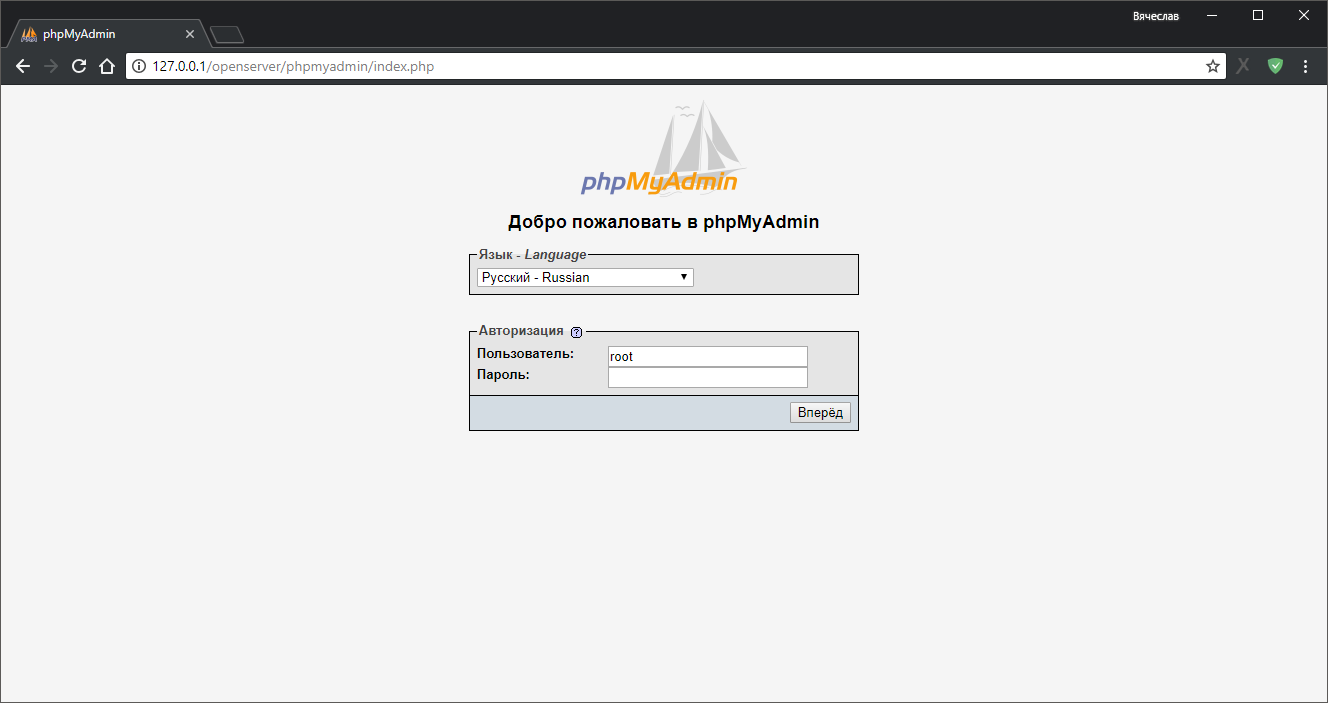


Рисунок 2 – Окно входа в phpMyAdmin

1. После авторизации видим веб-интерфейс программы. Нажмем на «Создать БД» в левой стороне веб-интерфейса программы. Назовем базу данных «pzcm», кодировку оставим как есть (рисунок 3).

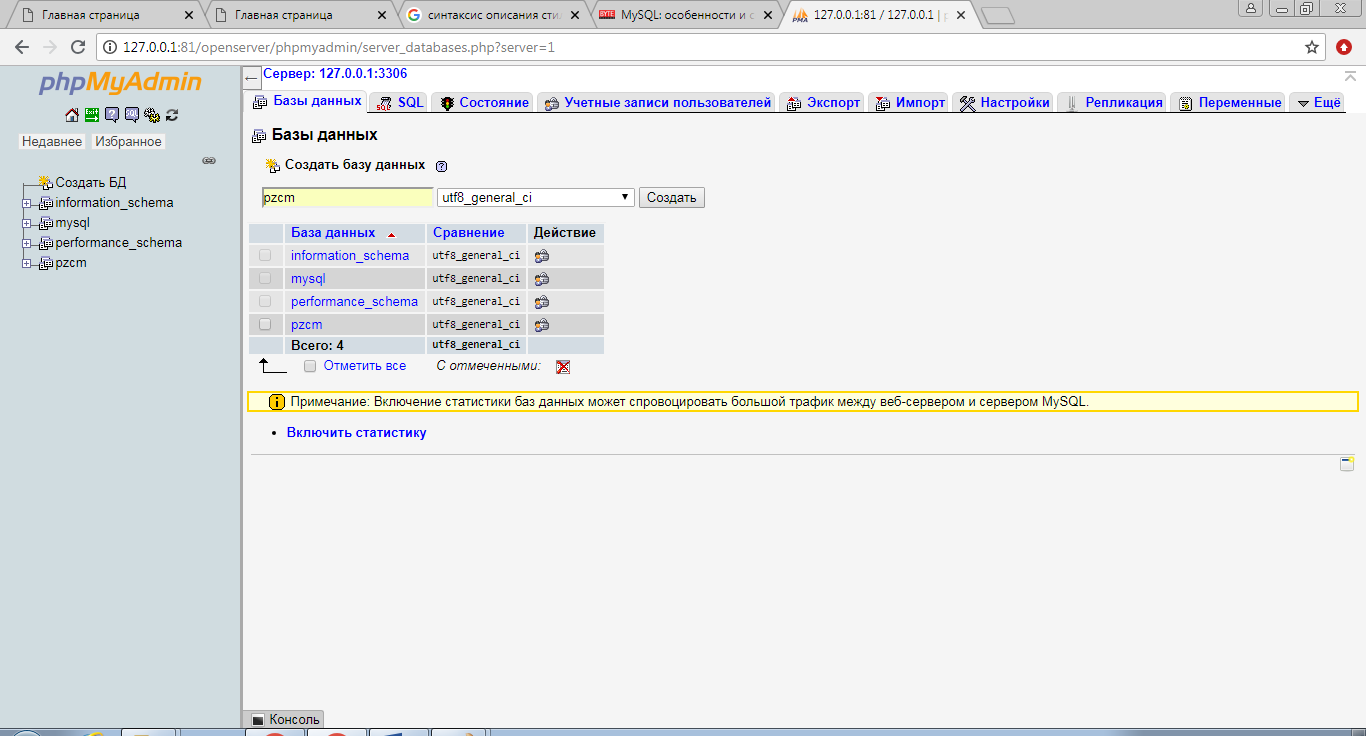


Рисунок 3 – Создание БД

1. Следующим шагом станет непосредственно импорт файла pzcm.sql в созданную БД. В левой колонке выберем созданную на прошлом шаге БД (pzcm) и справа вверху нажмем импорт (рисунок 4).

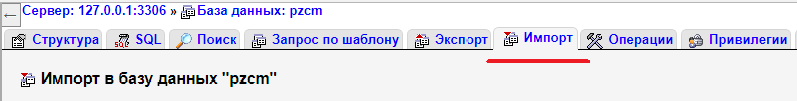


Рисунок 4 – Импорт в БД

1. Нажав на «Импорт», видим страницу «Импорт на текущий сервер» (Рисунок 5), нажимаем на кнопку «Выберите файл». Файл pzcm.sql находится в папке «site PZCM» папки «Курсовая работа» (…\Курсовая работа\site PZCM\DB).

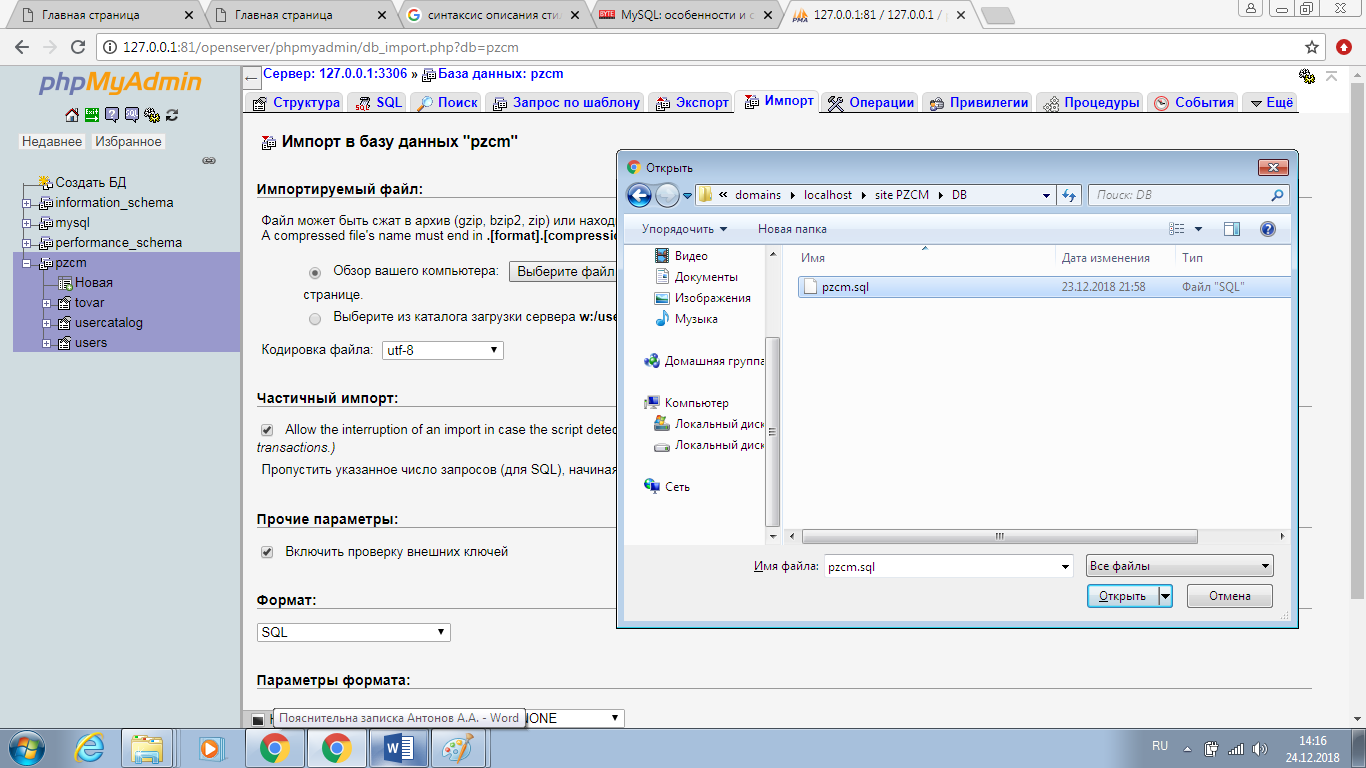


Рисунок 5 – Импорт в БД

1. После этого нажмем «Вперед» внизу страницы, и увидим сообщение о том, что импорт успешно завершен .
2. Отправим созданный сайт в папку …\OSPanel\domains и запустим его.
3. По завершению данных действий сайт полностью развернут и готов к использованию.

Лист регистрации изменений

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Номера листов (страниц) | | | | | Всего  листов (страниц) в докумен-те | № документа | Входящий  номер сопроводи-тельного документа | Подпись | Дата |
| Изм | Изменён-ных | Заменён-ных | Новых | Аннули-рован-ных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |