|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

**КУРСОВАЯ РАБОТА**

по дисциплине «Разработка программных приложений»

**Тема курсовой работы** Информационно-справочная система  
"Автомобили автосалона"

**Студент группы** ИВБО-05-17 Кувшинов Владислав Леонидович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

**Руководитель курсовой работы** доцент Синицын И.В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Работа представлена к защите «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2018 г.

Допущен к защите «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2018 г.

Москва 2018

|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Математического обеспечения и стандартизации информационных технологий

Утверждаю

Заведующий кафедрой МОСИТ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Головин С.А.

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2018 г.

**ЗАДАНИЕ**

**на выполнение курсовой работы по дисциплине**

«Разработка программных приложений»

Студент Кувшинов Владислав Леонидович Группа ИВБО-05-17

**Тема работы:** Информационно-справочная система «Автомобили автосалона»

**Исходные данные:**

**З**адание на курсовую работу – тема №20 Разработать информационно-справочную систему «Автомобили автосалона».

**Функционал разрабатываемой системы должен** предоставлять возможности полностью интерактивной системы с понятным дружественным графическим интерфейсом и обеспечивать необходимую функциональность, которая формируется в зависмости от заданной темы и предметной области изучаемых вопросов.

**Перечень вопросов, подлежащих разработке, и обязательного графического материала:**

Создание дистрибутива общего программного обеспечения (Реализация мультизагрузочного мобильного носителя с дистрибутивами операционных систем MS Windows 10 x64, Linux Mint 19 Mate x64, portable версий данных операционных систем, программных комплеков восстановления и сохранения, программных комплексов диагностики и системного сопровождения).

Освоение применения мультизагрузчиков Grub и Grub-2, освоение языков программирования для формирования специальных скриптов для управления Grub b Grub-2.

Установка и настройка MS Windows и Linux. В том числе настройка интерфейса с применением 3D столов типа Куб и доккетной линейки быстрого запуска приложений, практическое освоение командных языков платформы MS Windows и Linux(конфигурационное программирование).

Установка и настройка Web сервера Apache 2.x, FileZilla FTP Server 0.9.x, почтового сервера Mercury Mail Transport System 4.x, СУБД MySQL 5.x, интерпретаторы языков WEB программирования PHP 5.x и Perl 7.x, утилита обеспечения графического интерфейса при работе с СУБД MySQL.

Изучение жизненного цикла программ и создание программного комплекса с применением языков программирования PHP, PERL, JAVA и т.п., согласно темы курсового проектирования.

Тестирование и диагностика созданного программного продукта.

Портирование написанной программной системы на внешних хостах в сети Интернет.

Отчет по курсовой работе в виде расчетно-пояснительной записки .

**Срок представления к защите курсовой работы:** до «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2018 г.

**Задание на курсовую работу выдал** \_\_\_\_\_\_\_\_\_\_\_\_ Синицын И.В.

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2018 г.

**Задание на курсовую работу получил**  \_\_\_\_\_\_\_\_\_\_\_\_ Кувшинов В.Л.

Оглавление

[**ВВЕДЕНИЕ** 5](#_Toc535626797)

[**Наименование программы** 5](#_Toc535626798)

[**Функциональное и эксплуатационное назначение** 5](#_Toc535626799)

[**Актуальность** 5](#_Toc535626800)

[**Описание и обоснование метода выбора программных средств** 5](#_Toc535626801)

[**Постановка задачи** 5](#_Toc535626802)

[**ОСНОВНАЯ ЧАСТЬ. Установка и настройка операционной системы** 6](#_Toc535626803)

[**Подготовка дистрибутива для установки ОС Windows** 6](#_Toc535626804)

[**Установка Windows 7** 7](#_Toc535626805)

[**Создание загрузочной флешки с Ubuntu в OC Windows** 9](#_Toc535626806)

[**Создание загрузочной флешки с Ubuntu в OC Ubuntu** 9](#_Toc535626807)

[**Установка операционной системы Ubuntu** 10](#_Toc535626808)

[**Установка и настройка программного комплекса XAMPP** 12](#_Toc535626809)

[**СТРУКТУРА И ОПИСАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ.** 13](#_Toc535626810)

[**Базы данных** 13](#_Toc535626811)

[**Описание информационной системы. Функционал** 14](#_Toc535626812)

[**Размещение информационной системы на хостинге** 21](#_Toc535626813)

[**КОД ПРОГРАММЫ** 21](#_Toc535626814)

[**ЗАКЛЮЧЕНИЕ** 34](#_Toc535626815)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 34](#_Toc535626816)

# 

# **ВВЕДЕНИЕ**

## **Наименование программы**

Информационно-справочная система «Автомобили автосалона» – программа, предназначенная для получения пользователем информации о предложениях автосервиса с возможностью бронирования понравившейся позиции из каталога.

## **Функциональное и эксплуатационное назначение**

Система имеет такие функции как: вывод информации об автомобилях и их ценах, ввод пользователем новых предложений, редактирование или удаление существующих.

## **Актуальность**

Данная программа может быть использована на сайтах по продаже автомобилей, предоставляющих пользователям возможность ознакомиться и приобрести понравившееся предложение. Также существует возможность улучшения функционала программы (например, добавление фильтров для поиска информации в таблице).

## **Описание и обоснование метода выбора программных средств**

Для успешной работы программы на локальном сервере (localhost) на компьютере должны быть установлены операционная система Windows и программное обеспечение XAMPP. Также программный комплекс может быть использован в сети Интернет с помощью перехода на сервер (файлы программы хранятся на бесплатном хостинге). Для разработки программного продукта были использованы языки программирования PHP, MYSQL, HTML/CSS.

## **Постановка задачи**

При разработке программного продукта был сформулирован ряд определенных задач, которые были успешно выполнены в процессе разработки:

* Разработать удобное и понятное для пользователя приложение для получения информации об автомобилях и их ценах.
* Использовать базы данных.
* Добавить в функционал программы такие функции, как:

- авторизация в программу через логин и пароль;

- регистрация нового пользователя в программе;

- получение информации в таблицах;

- добавление необходимой информации вручную;

- удаление ненужной пользователю информации (также вручную).

Важно отметить, что пользователь не может ввести неверные данные в поля для ввода, например, буквы в числовые поля. Также недоступен ввод пустых полей.

Пользователи, не авторизованные в программе, не смогут получить к ней доступ. (данная механизм реализован через PHP-сессии).

## **ОСНОВНАЯ ЧАСТЬ. Установка и настройка операционной системы (с программным обеспечением)**

## **Подготовка дистрибутива для установки ОС Windows**

Дистрибутив операционной системы можно разместить на USB-флеш-накопителе или на оптическом диске. Первый вариант считается более удобным и современным, поэтому далее будет рассказано именно о нем.

Необходимо иметь флеш-накопитель достаточного объема, который способный вместить дистрибутив операционной системы Windows. В данном конкретном случаем будет создаваться установочная флешка для Windows 7, так что объем накопителя должен быть как минимум 4 гигабайта.

   Существует несколько способов создания загрузочной флешки, которые в основном можно разделить по классификации:

* Ручной
* Автоматический

Далее идет пояснение, как использовать автоматический способ, дабы продемонстрировать то, с чем может столкнуться пользователь при желании самостоятельно обновить свою операционную систему. Автоматический способ подразумевает под собой использование программного обеспечения, превращающего обычную, стандартную флешку в средство загрузки и установки операционной системы.

На рынке программного обеспечения существует множество программ с подобным функционалом, которые в свою очередь делятся на платные и бесплатные, узкопрофильные и широкопрофильные.

Примером условно-бесплатного программного обеспечения широкого профиля является программа UltraISO, в функционал которой входят поддержка более 30 различных форматов и запись на разные виды носителей. В данном случае подобный функционал не является обязательным к применению, поэтому мы воспользуемся бесплатным программным обеспечением, предоставляемым компаний Microsoft для создания дистрибутивов их же собственной операционной системы - Windows USB/DVD Download Tool.

Зайдя на официальный сайт Microsoft и найдя страницу с описанием этой программы, можно произвести загрузку программного обеспечения и последующую установку согласно прилагаемой инструкции на официальном сайте.

Установив программу Windows USB/DVD Download Tool, далее следует получить образ операционной системы Windows. Образ операционной системы Windows с расширением .iso, который необходим для работы с программой можно приобрести в официальном магазине Microsoft. Либо, если у пользователя уже имеется приобретенный ключ Windows, то, введя его на официальном сайте, можно так же скачать персональный образ операционной системы.

После того как образ операционной системы был куплен и успешно загружен на жесткий диск, необходимо вставить флеш накопитель в USB-порт, произвести запуск программы Windows USB/DVD Download Tool и указать путь к скачанному образу ОС Windows.

Когда все первоначальные приготовления выполнены, флеш накопитель успешно присоединен к компьютеру и в программе Windows USB/DVD Download Tool указан путь к образу операционной системы, необходимо продолжить процесс, нажав на кнопку «Далее».

На следующем диалоговом окне необходимо сделать выбор между носителями для создания и записи установочного образа. Так как в конкретном случае пользователь работает с на USB-флеш-накопителем, а не оптическим диском, то следует выбрать первый вариант.

Следующее диалоговое окно предоставляет пользователю выбор из накопителей, которые в данный момент подключены к компьютеру, чтобы мы выбрать тот, на который будет записан дистрибутив. В выпадающем меню пользователь выбирает нужный USB-флеш-накопитель. Необходимо отметить, что все данные на накопителе будут уничтожены после создания дистрибутива, поэтому следует предпринять определенные действия для переноса важных данных на другой носитель, прежде чем продолжать процедуру создания дистрибутива.

После выполнения описанных выше шагов следует произвести нажатие на кнопку «Начать запись». Программа начнет создание установочного образа операционной системы на флеш накопителе. О процессе установки операционной системы будет рассказано ниже.

## **Установка Windows 7**

Установка операционной системы Windows 7 на компьютер начинается после создания дистрибутива этой системы. Перед началом установки необходимо зайти в BIOS/UEFI и настроить очередность запуска запоминающих устройств, установив флэш-накопитель на первое место. Это делается для того, чтобы при загрузке компьютера происходила именно с флэш-накопителя.

После перезагрузки компьютера запускается утилита по установке Windows на компьютер. Самое первое окно предлагает пользователю выбрать язык системы, формат времени и раскладку клавиатуры. Далее будет предложено ознакомиться с условиями лицензионного соглашения. После успешного прочтения и подтверждения согласия с правилами продолжается установка. Далее выбирается тип установки: обновление или полная установка. При установке системы «с нуля» выбирается вариант полной установки и идет переходим на окно настройки разделов жесткого диска. На этом этапе пользователь может разделить физический диск компьютера на несколько логических разделов с определенными размерами или оставить один раздел, в зависимости от своих предпочтений и от требований выполняемых задач. После этого можно начать установку операционной системы.

После того, как Windows установился нам предложено создать пользователя. Мы создаем его задав имя пользователя и имя компьютера, а после и пароль.  Следующим этапом установки будет активация версии Windows. Это можно сделать, например, с помощью лицензионного ключа, если таковой имеется. (При первом подключении к Интернету желательно иметь активированную версию Windows). Однако, если у вас нет лицензионного ключа (кода), можно пропустить этап активации и нажать кнопку "далее".  Начнется настройка операционной системы, но вам нужно будет активировать вашу версию Windows в течение одного месяца. На следующем этапе пользователю будет предложено принять рекомендуемые параметры и настройки, такие как установка важных и рекомендуемых обновлений, обеспечение более безопасного обзора Интернета, и так далее. В этом случае лучше выбрать пункт "Отложить решение". После этого пользователю следует сверить и настроить параметры даты и времени.

В случае если Windows сам обнаружит и успешно установит сетевые адаптеры, пользователю будет предложено выбрать тип сети. Нужно ознакомиться с предложенными вариантами и выбрать наиболее подходящий.

После этого этапа установка Windows завершена. Теперь мы можем перейти к настройке системы. Для корректной работы нам понадобиться настроить сеть. Основная настройка сети происходит в центре управления сетями и общим доступом. Здесь можно поменять сетевое расположение, если во время установки вы выбрали неправильный вариант, или ваша сеть была изменена. Сетевое расположение компьютера важно, так как от его настройки зависит настройка строгости брандмауэра.

Так же в этом окне можно просмотреть карту сети - это графическое представление расположения компьютеров и устройств в локальной сети.  Набор данных, необходимых пользователю для подключения компьютера к интернету или к локальной сети, можно увидеть в окне "Сетевые подключения".



Пример карты сети

Попасть на данную страницу можно из окна "Центр управления сетями и общим доступом", перейдя по ссылке "Изменение параметров адаптера". Чтобы получить информацию о сетевом подключении, следует выбрать опцию "свойства" нужного адаптера. В случае с Windows 7 в окне свойств адаптера нужно выбрать "Протокол Интернета версии 4 (TCP/IPv4)".

Здесь же пользователь может настроить шлюзы, DNS сервера, IP-адреса и т.д., получив необходимые сведения у своего провайдера.

Используя окно «Состояние сети» можно не только просмотреть любые данные о сети подключения, но и узнать такие детали, как, например, MAC-адрес или IP-адрес.

Если возникает ситуация с непредвиденными сбоями в работе сети, то одним из способов их возможного устранения является диагностика подключения ("Устранение неполадок"). Оно анализирует состояние подключения и может предложить пользователю данные о неисправностях и способы их устранения.

## **Создание загрузочной флешки с Ubuntu в OC Windows**

Для начала пользователю необходимо скачать приложение Unetbootin для Windows с официального сайта Ubuntu. После успешной загрузки приложения следует запустить Unetbootin с правами администратора.

При запуске приложения пользователю предложат выбрать дистрибутив для установки системы. В данном пункте необходимо указать пункт “Ubuntu”.

Далее в пункте “Выберите версию” пользователь указывает необходимую ему версию. В следующем пункте “Тип диска” указать нужный параметр - “устройство USB”, а в пункте “Носитель” выбрать свою установочную флешку. Для завершения процесса создания дистрибутива необходимо нажать на кнопку “ОК”.

## **Создание загрузочной флешки с Ubuntu в OC Ubuntu**

1) Для установки Ubuntu флешка должна быть не менее 1 Гб. Также нужно иметь ISO образ Ubuntu.

2) Для создания установочной флешки нам понадобится программа Gрartеd. Установить его Вы можете с помощью вашего менеджера пакетов или использовать команду: ***sudo apt-get install Gрartеd***.

GPartеd будет установлен в **Systеm > Аdministratiоn > Pаrtitiоn Editоr**.

3) GPartеd отображает список дисков. Теперь, всё что нужно — это отформатировать флешку и сделать её загрузочной.

Важное замечание: после форматирования все данные на USB флешке будут уничтожены, поэтому пользователю следует заранее позаботиться об их сохранении.

4) После подключения флешки следует выбрать пункт меню GPаrtеd> «Обновить устройства». После обновления в списке отобразится нужный USB.

5) Перед тем как форматировать, нужно ее отсоединить, нажав правой кнопкой мыши имени устройства, затем «отсоединить».

6) Далее следует отформатировать флешку, нажав правой кнопкой на имя флешки и выбрав пункт “Форматировать” в Fаt32. Для лучшей совместимости можно выбрать FАT16.

7) В открывшемся окне в нижней части GParted пользователь может поставить в очередь задания, которые затем будут последовательно выполнены. При нажатии правым кликом на имени диска в GPartеd и выборе пункта меню 'Manage Flags' следует убедиться, является ли данный диск загрузочным.

8) Если он таковым не является следует выбрать нужный пункт, нажав на галочку и затем завершить работу с GParted. После произведенных манипуляций флешка готова для установки Ubuntu. Далее пользователю следует отсоединить флешку и вставить ее заново для перемонтирования, которое может понадобиться в дальнейшем.

9) Скачиваем Unetbootin с официального сайта Ubuntu.

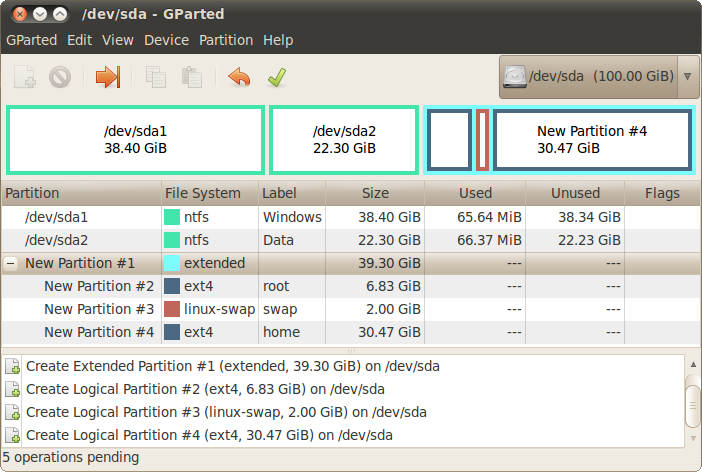
10) Unetbootin будет установлен в категорию Приложения > Системные. Откройте Unetbootin.

11) В верхней части экрана Unetbootin при нажатии на иконку 'Diskimage', необходимо загрузить ISO-образ. Выбрав нужный накопитель, можно начать копирование.

## **Установка операционной системы Ubuntu**

Прежде чем приступить к установке Ubuntu необходимо выполнить разметку диска. Для этого используем встроенную по умолчанию в Ubuntu утилиту Gparted. Как известно, существующего объема на диске часто не хватает. Для компенсации этого недостатка используется так называемый механизм подкачки, использующий определенный раздел на диске, называемый свопом. Кроме свопа, для работы системы нам безусловно потребуется раздел для системных и пользовательских файлов.

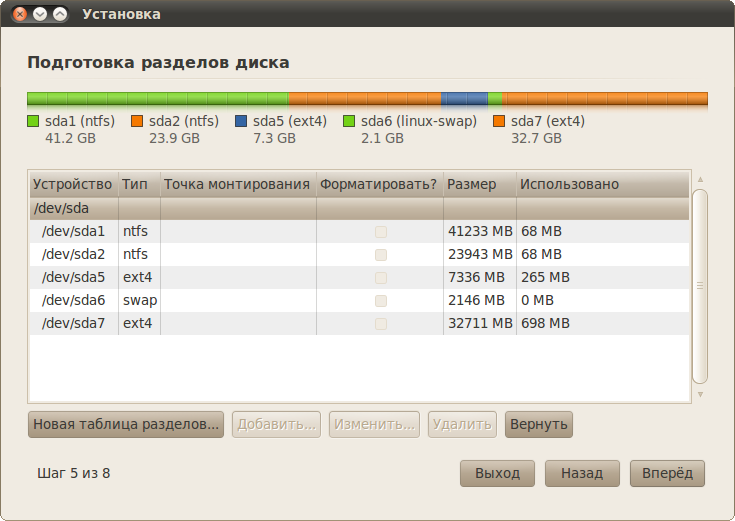
Мы будем использовать стандартную разметку.  Это значит, что мы создадим один раздел под систему,  второй под данные пользователя и еще один под своп. Сперва создадим один расширенный раздел. Этот раздел мы и разобьём на несколько частей. В нашем случае это будет три. Для создания раздела нажимаем правой кнопкой мыши на любое свободное место, выбираем в появившемся меню пункт NEW. Итак, новый раздел создан. Для создания подразделов повторяем алгоритм только правой кнопкой нажимаем на только что созданный расширенный раздел, снова выбираем NEW. При этом необходимо указать для него файловую систему ext4. Затем формируем своп.  Все оставшееся пространство отдадим для пользовательских файлов и данных. Для нее снова используем файловую систему ext4.



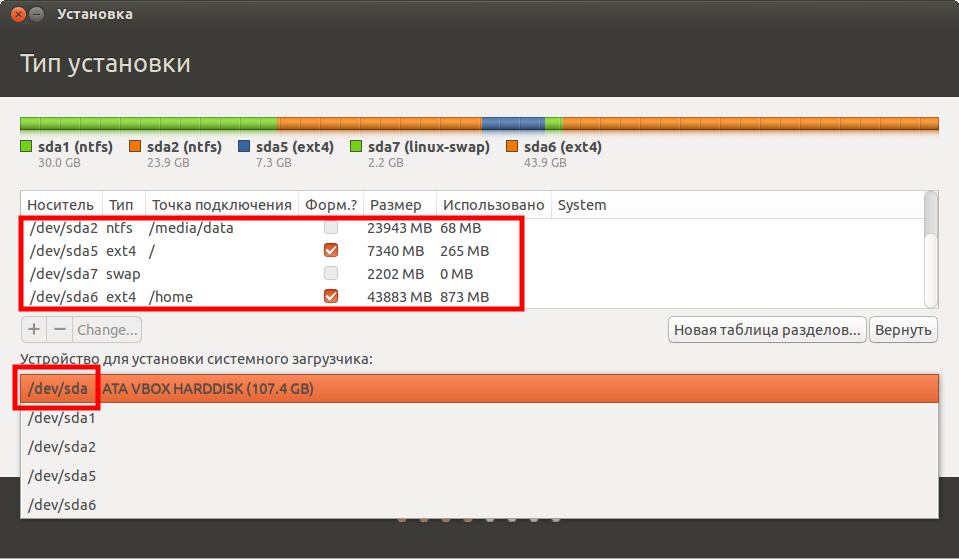
Для завершения процесса разметки нажимаем «Apply all operations».

Для того, чтобы начать установку Ubuntu нам необходимо перейти на рабочий стол запущенной с LiveCD Ubuntu, на котором в есть ярлык, называемый «Установить Ubuntu 10.04 LTS». Именно этот ярлык позволяет нам запустить процесс установки. На начальном этапе выбираем язык системы и часовой пояс, раскладку клавиатуры.

Вторым этапом укажем Ubuntu место для установки. Установщик же покажет нам графическое отображение винчестера, и соответственно предложит дальнейшие действия. Сперва необходимо разметить жесткого диска, поэтому мы выбираем третий пункт и нажимаем кнопку «Вперёд». Появится окно со списком разделов винчестера. Здесь мы видим все созданные нами в GParted разделы дисков и все оставшиеся названия дисков.



Теперь осталось только указать, какой диск с какой целью используется. Начнём с системного диска. При разметке винчестера мы назначили ему sda5. Далее необходимо открыть окно редактирования свойств. Так как sda5 будет системным или корневым разделом системы, поэтому в разделе, где указывается точка монтирования, нужно указать «/». Также, в разделе содержащем информацию о том, как он будет использоваться необходимо установить параметр «Журналируемая файловая система Ext4». Именно она чаще всего используется для корня. Чтобы завершить редактирование раздела необходимо поставить галочку в поле «Форматировать раздел».



Следующий раздел sda6. Для него действуем аналогично, единственным отличием будет то, что здесь надо указать точку монтирования «/home».

Таким образом для всех дисков, кроме свопа должны быть указаны точки монтирования.

Итак, основной этап завершен. Теперь система попросит выбрать часовой пояс. Сделать это можно несколькими способами. Например, введя название города. Также можно выбрать место на карте. Далее следует выбрать раскладку клавиатуры.

Для начала процесса установки необходимо ввести имя первого и основного пользователя для Ubuntu. Указанный пользователь будет считаться администратором. Он будет обладать правом полного доступа к управлению системой. Далее жмем кнопку «Продолжить» для начала процесса установки. Важно то, что отменить процесс установки после его запуска невозможно.

После успешной установки система предложит перезагрузиться. Перезагружаем компьютер, не забыв при этом обратно перенастроить BIOS на загрузку с жесткого диска, и мы попадаем в установленную Ubuntu. Дополнительных настроек система не требует, так как основные параметры были заданы на этапе установки.

## **Установка и настройка программного комплекса XAMPP**

I.Компоненты XAMPP:

* X – обозначает, что сборка будет работать на всех операционных системах;
* Apache – веб-сервер, поддерживающий основные версии ОС, гибко настраиваемый и надёжный;
* MySQL – одна из самых известных систем управления базами данных;
* PHP – язык программирования, поддерживаемый почти всеми хостинг-провайдерами и лидер среди языков, используемых для создания веб-сайтов;
* Perl – лёгкий в освоении язык программирования, который, к тому же, поддерживает модули сторонних разработчиков.

II. Основные этапы настройки:

1.Смена корневой директории

По умолчанию директория для localhost будет "c:\xampp\htdocs\", именно в ней находятся файлы, которые отображают в браузере вступительную страницу XAMPP. При необходимости ее можно изменить в настройках сервера Apache, но это делать не обязательно. Для изменения директории остановите сервер Apache, в файле C:\xampp\apache\conf\httpd.conf (этот файл отвечает за конфигурацию сервера) укажите путь к желаемой директории, где будет находиться корень сервера:

DocumentRoot "C:/xampp/htdocs" и <Directory "C:/xampp/htdocs">

Запустите сервер снова. Для проверки работы можно в этой директории создать какой-нибудь index.html или index.php файл и набрать в браузере адрес - http://localhost, должно загрузиться содержимое этого файла. Если вы изменили директорию, доступ к phpMyAdmin останется тот же: http://localhost/phpmyadmin

2.Установка уровня отображения ошибок для PHP

Изначально XAMPP сконфигурирован для разработчика, т.е. в браузере будут отображаться не только критические ошибки PHP, но и различного рода предупреждения, например, "Strict Standards: ...", "Notice: ..." и другие. Для изменения уровня отображения ошибок PHP нужно открыть файл "C:\xampp\php\php.ini" в любом редакторе и в нем поставить значение:

- error\_reporting = E\_ALL

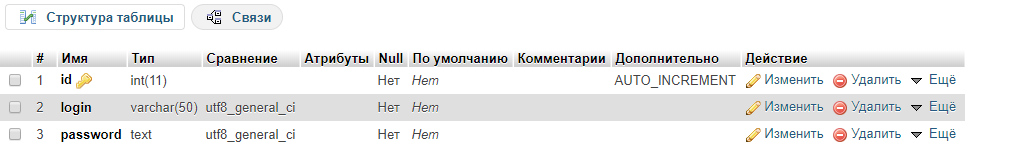
При этом значении будут отображаться только критические ошибки PHP.

После внесения изменений в файл конфигурации PHP (php.ini) так же нужно перезагружать сервер Apache. В контрольной панели XAMPP напротив "Apache" нажать кнопку "Stop", если сервис уже запущен, дальше после остановки нажать кнопку "Start".

# **СТРУКТУРА И ОПИСАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ.**

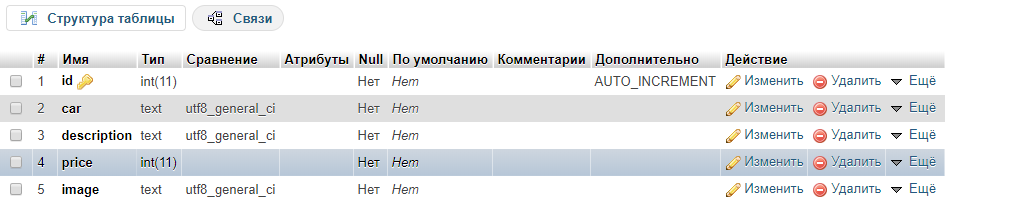
## **Базы данных**

При проектировании информационно-справочной системы была создана база данных autoservice. Были созданы 3 таблицы, в users было решено хранить данные о пользователях информационной системы, в cars\_list – сведения об автомобилях, в orders – сведения о заказах. Поля таблицы users: id (идентификационный номер пользователя – является также первичным ключом с параметром AUTO\_INCREMENT), login (собственно, логин пользователя), password (пароль пользователя).



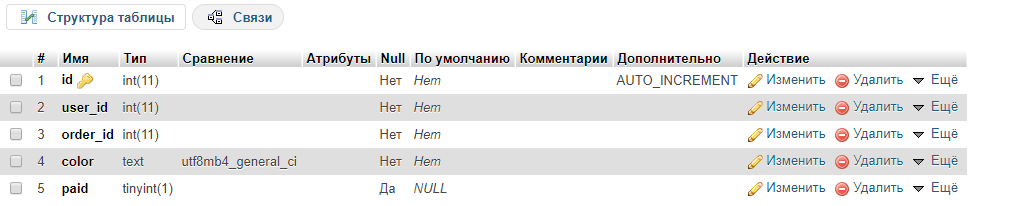
(структура таблицы users)

В таблице cars\_list было определено 5 полей: id (идентификационный номер команды – первичный ключ с параметром AUTO\_INCREMENT), car (Название автомобиля), description (Описание автомобиля), price (Стоимость автомобиля), image (Путь к изображению автомобиля).



(структура таблицы cars\_list)

В таблице orders было определено 5 полей: id (идентификационный номер команды – первичный ключ с параметром AUTO\_INCREMENT), user\_id (id Заказчика), order\_id (id Заказа), color (Выбранный цвет), paid (Состояние бронирования).

  
(структура таблицы orders)

## **Описание информационной системы. Функционал**

При использовании системы в первый раз открывается страница авторизации (index.php). Пользователю предлагается ввести свой логин и пароль (если таковые имеются), чтобы войти в информационно-справочную систему.

В начале файла index.php через php-код создается сессия. Проверки через сессии сделаны для того, чтобы незарегистрированные и неавторизованные пользователи не смогли получить доступ к информационной системе.

<?php session\_start(); ?>

После ввода пользователем своего логина и пароля идет попытка подключения к базе данных tours.

<?php

$db = mysqli\_connect ("localhost","root","","autoservice");

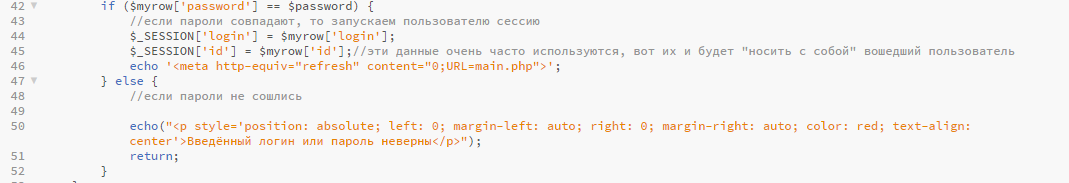
?>

После подключения к базе данных выполняется запрос на выборку данных о логинах и паролях всех пользователей.

$result = mysqli\_query($db, "SELECT \* FROM users WHERE login='$login'");

$myrow = mysqli\_fetch\_array($result);

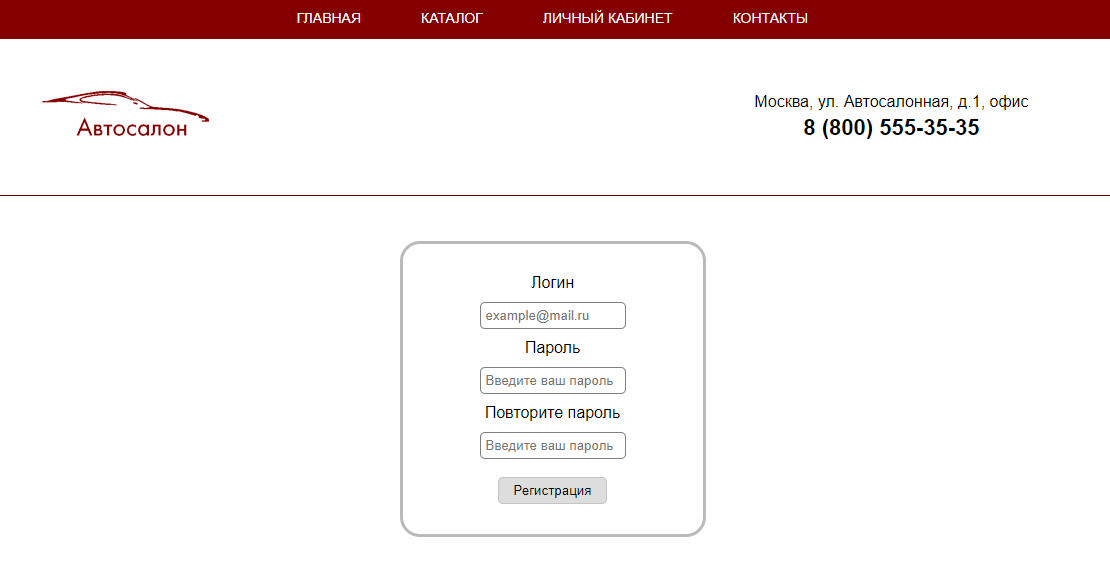
Далее в php коде проверяется совпадения логина и пароля (одновременно) с любой из строк выполненного ранее запроса. При совпадении пользователь считается авторизованным и происходит переход на страницу главного меню (main.php). Также в переменную $\_SESSION['login'] записывается значение для последующей проверки его наличия в других php-файлах.

Если совпадений при вводе логина и пароля не обнаружено, пользователю будет показано соответствующее сообщение.

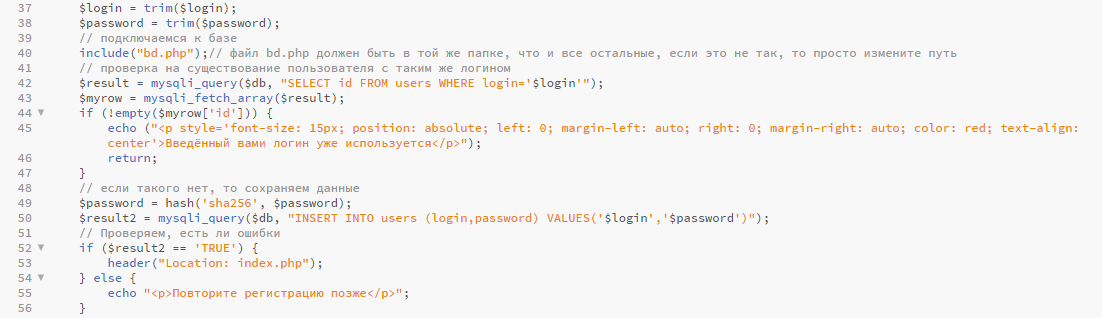
Неавторизованный пользователь может совершить попытку войти в систему напрямую через другие php-файлы, а не через авторизацию с помощью логина и пароля. Для предотвращения данных действий и отказа в доступе в каждом php-файле идет проверка на наличие значения переменной $\_SESSION['login']. Выше было отмечено, что в файле index.php при успешном вводе логина и пароля данная переменная принимает значение введённого имени. Соответственно, если авторизация не была пройдена пользователем, то у данной переменной не будет указано значение.

В таком случае неавторизованный пользователь не получит доступ к системе и будет возвращён на страницу авторизации.

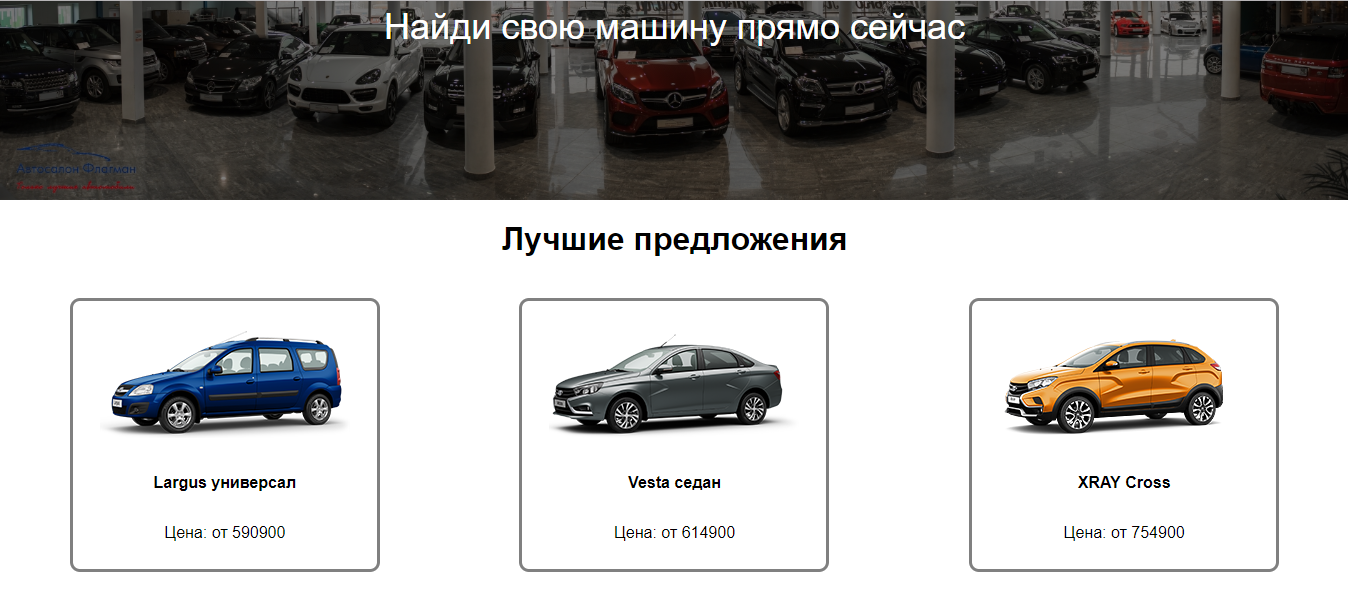
Рассмотрим страницу регистрации, на которую новому пользователю предлагается перейти, если еще не создана учетная запись. В ней можно заполнить новый логин и новый пароль, которые впоследствии будут использоваться для входа в систему. При нажатии кнопки «Зарегистрироваться» данные о новом пользователе будут переданы и записаны в таблицу users.



(регистрация –reg.php)



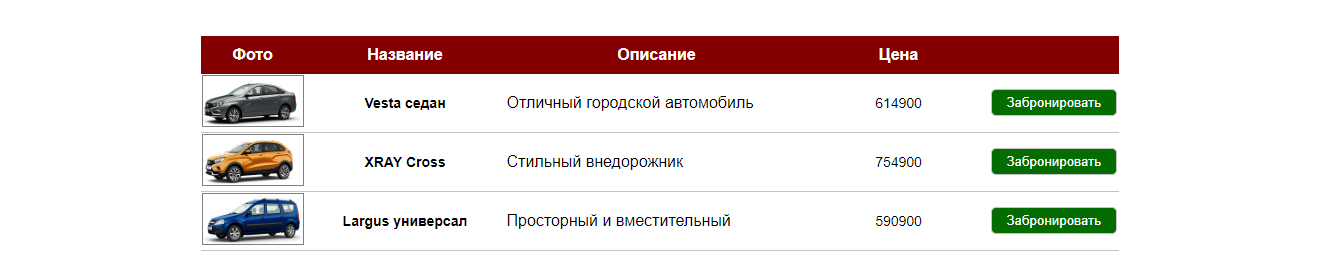
После того, как пользователь верно введет свой логин и пароль на странице авторизации (идет проверка совпадение логина и пароля с любой из строк таблицы users), будет произведен вход в систему – в главное меню.



(главное меню – main.php)

Из главного меню пользователь может получить информацию о наиболее выгодных предложениях, а также имеет возможность перейти к просмотру любой из страниц сайта.

При выборе вкладки «Каталог» пользователь будет направлен на страницу, содержащую список всех доступных предложений на ресурсе.



(список автомобилей – catalog.php)

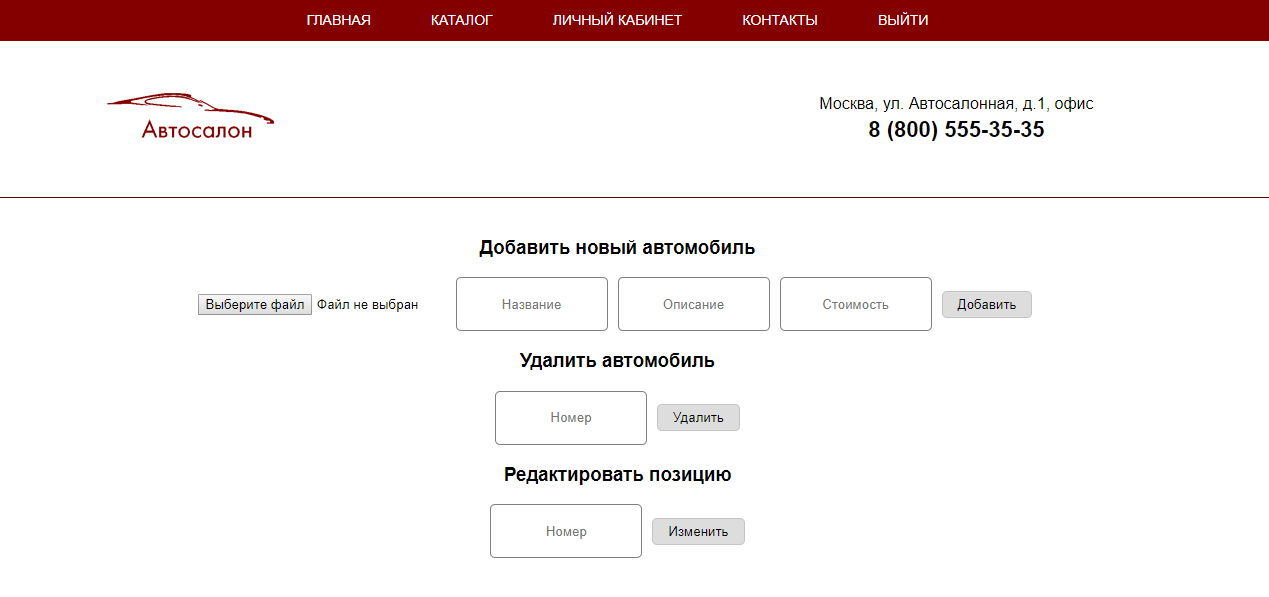
Здесь же пользователь может забронировать понравившийся автомобиль, нажав на кнопку «Забронировать» напротив нужного предложения.

Порядок работы php-кода и заполнения таблицы каталога заключается в построчном выводе информации из «cars\_list» в форму html-таблицы.



Стоит обратить внимание на то, что на представленном выше скриншоте показан лишь пример заполнения таблицы. Редактор и администратор вправе изменять содержимое данных списков (таблиц), добавляя новые или удаляя текущие строки.

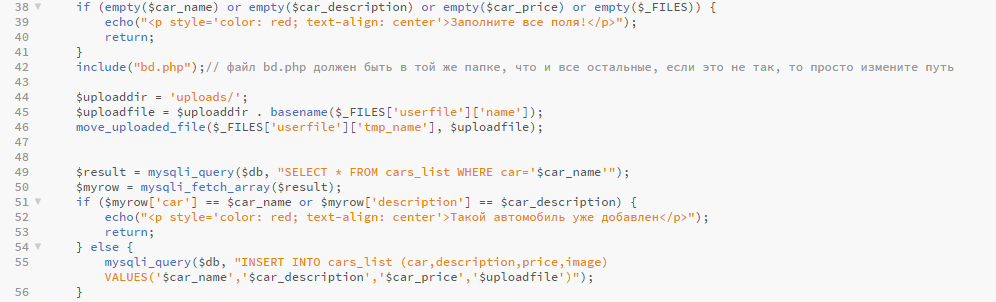
Во время загрузки на страницу catalog.php происходит сравнение логина текущего пользователя (с помощью переменной $\_SESSION[‘login’]) с логином редактора или администратора. В случае совпадения, пользователю открывается расширенный функционал страницы, позволяющий добавлять, удалять, и редактировать записи.



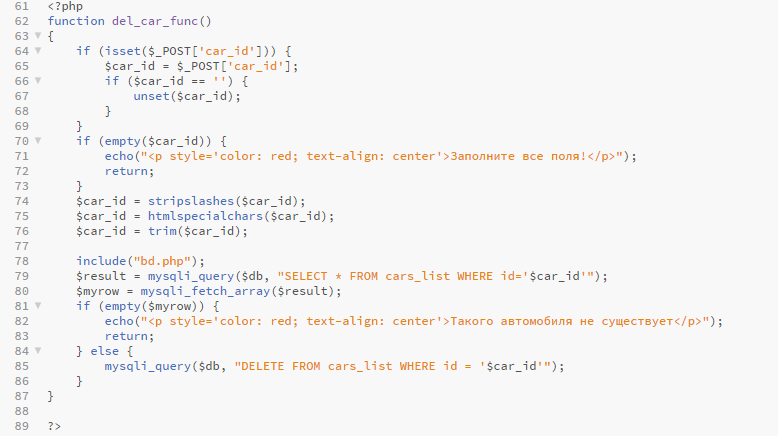
(панель администратора – catalog.php)

При нажатии на кнопку «добавить» информация, находящаяся в полях ввода будет загружена в соответствующие столбцы таблицы «cars\_list». Ограничения при вводе:

* все поля должны быть заполнены (не допускается добавление информации хотя бы при одном пустом поле);
* поле «стоимость» должно быть числовым (в противном случае пользователь будет оповещен о неверном вводе);

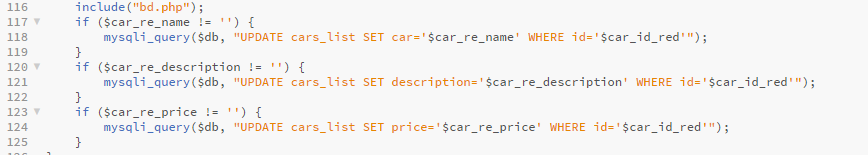


Для удаления информации о туре, пользователь использовать форму, обозначенную как «Удалить тур». Рассмотрим работу формы удаления. PHP-код соединяется с базой данных tours и делает запрос на выборку всех значений из таблицы «cars\_list» с полем «id», совпадающим с введённым значениям, затем полученная строка удаляется из таблицы.



Чтобы увидеть результат удаления в действии, пользователь может нажать на кнопку «обновить» и удаленная строка исчезнет из каталога.

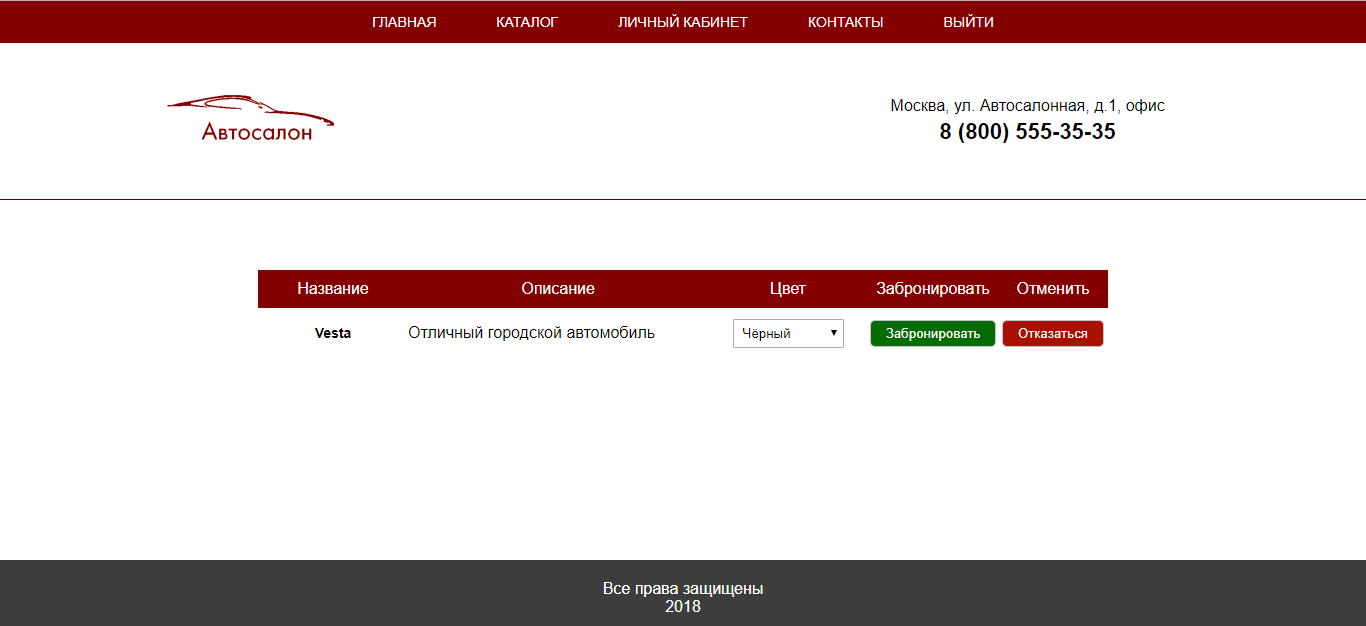
Работа функции «Редактировать тур» отличается от вышеизложенных sql запросом UPDATE.



При окончании работы с системой пользователь может выйти из нее, нажав на соответствующую кнопку вверху на навигационной панели. При нажатии на кнопку «Выйти» в файле exit.php активируется скрипт, который «уничтожает (разрушает)» все данные, ассоциированные с текущей сессией и происходит переход на страницу авторизации (index.php).



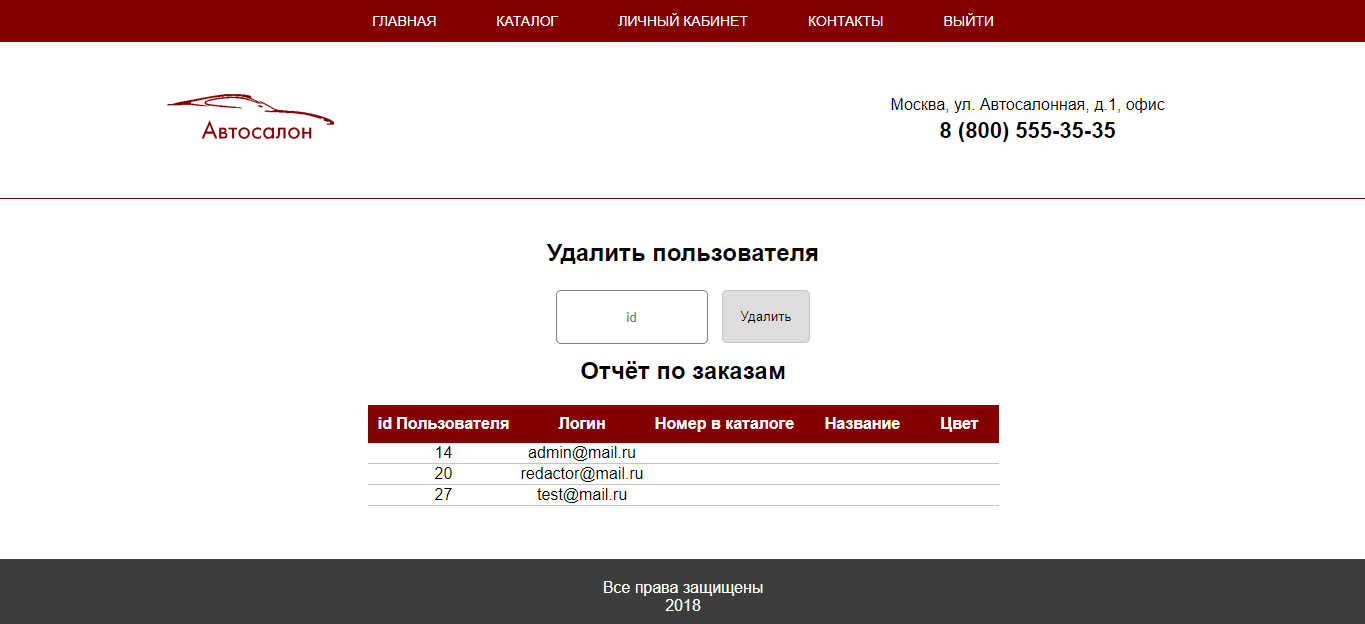
Следующая вкладка навигационной панели – «Личный кабинет». Он так же имеет различное представление для пользователей с разными полномочиями. Пользовательский вид содержит список заказов, форму выбора цвета, кнопки отмены и бронирования.



(Личный кабинет пользователя – cabinet.php)

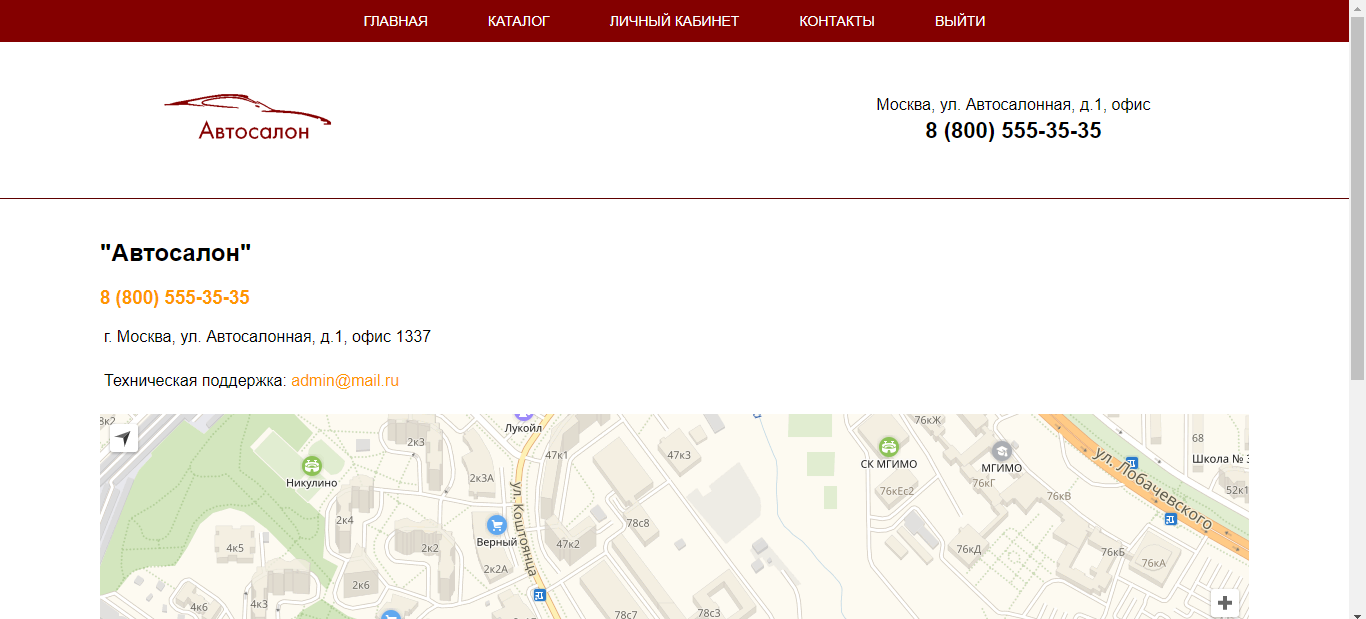
При нажатии на кнопку бронирования, происходит утверждение заказа. Кнопка отмены удаляет заказ.

Администратор на той же странице имеет форму для удаления пользователей (механизм аналогичен процедуре удаления туров) и список заказов пользователей.



(Личный кабинет администратора – cabinet.php)

Вкладка «Контакты» содержит в себе информацию о компании, адрес, контактный телефон, карту.



## **Размещение информационной системы на хостинге**

Для доступа к информационно-справочной системе в Интернете было решено разместить всех необходимые для работы программы файлы на бесплатном хостинге (000webhost.com). После создания аккаунта на сайте был получен доступ к серверу. Для хранения файлов на сервере была создана пустая база данных, в которую позже были импортированы созданные ранее на локальном сервере базы данных. Размещение файлов на хостинг было произведено с помощью FTP-клиента FileZilla.

# **КОД ПРОГРАММЫ**

Index.php

**<?php** session\_start(); **?>**<!DOCTYPE html>  
<html lang="en">  
  
<head>  
 <meta charset="UTF-8">  
 <title>Улётный Тур - Вход</title>  
 <link rel="shortcut icon" href="images/favicon.ico" type="image/x-icon">  
 <link rel="stylesheet" href="style.css">  
</head>  
  
<body>  
  
<div class="navbar">  
 <div class="header\_items header\_item\_1">  
 <a href="main.php">Главная</a>  
 </div>  
 <div class="header\_items header\_item\_2">  
 <a href="catalog.php">Каталог</a>  
 </div>  
 <div class="header\_items header\_item\_3">  
 <a href="cabinet.php">Личный кабинет</a>  
 </div>  
 <div class="header\_items header\_item\_4">  
 <a href="contacts.php">Контакты</a>  
 </div>  
 **<?php  
 if** (!**empty**($\_SESSION['login'])) {  
 **echo** ("<form class='header\_items' method='post' action='exit.php'>  
<input type='submit' name='exit' value='Выйти'>  
</form>");  
 }  
 **?>**</div>  
  
<header>  
 <div class="logo">  
 <img alt="Улётный Тур" src="images/logotype.png">  
 </div>  
 <div class="cabinet\_access">  
  
 </div>  
 <div class="contacts">  
 <p style="text-align: center;">Москва, ул. Туристическая, д.1, офис</p>  
 <p style="font-size:22px; font-weight: bold; text-align: center;">8 (800) 555-35-35</p>  
 </div>  
</header>  
  
<div class="authorization\_main\_hor">  
 <div class="authorization\_main\_ver">  
 <form method="post" class="authorization">  
 <p>Логин</p>  
 <input type="email" class="form\_input" placeholder="example@mail.ru" name="login" size="15" maxlength="50">  
 <p>Пароль</p>  
 <input type="password" class="form\_input" placeholder="Введите ваш пароль" name="password" type="password" size="15"  
 maxlength="15">  
 <br>  
 <input type="submit" class="log\_button" name="log\_btn" value="Вход">  
 <br>  
 <input type="submit" class="log\_button" name="reg\_btn" value="Регистрация">  
 </form>  
  
  
 **<?php  
 if** (**isset**($\_POST['reg\_btn'])) {  
 header('Location: reg.php');  
 }  
 **if** (**isset**($\_POST['log\_btn'])) {  
 **if** (**isset**($\_POST['login'])) {  
 $login = $\_POST['login'];  
 **if** ($login == '') {  
 **unset**($login);  
 }  
 } //заносим введенный пользователем логин в переменную $login, если он пустой, то уничтожаем переменную  
 **if** (**isset**($\_POST['password'])) {  
 $password = $\_POST['password'];  
 **if** ($password == '') {  
 **unset**($password);  
 }  
 }  
 //заносим введенный пользователем пароль в переменную $password, если он пустой, то уничтожаем переменную  
 **if** (**empty**($login) **or empty**($password)) //если пользователь не ввел логин или пароль, то выдаем ошибку и останавливаем скрипт  
 {  
 **exit** ("<p style='color: red; text-align: center'>Заполните все поля!</p>");  
 }  
 //если логин и пароль введены,то обрабатываем их, чтобы теги и скрипты не работали, мало ли что люди могут ввести  
 $login = stripslashes($login);  
 $login = htmlspecialchars($login);  
 $password = stripslashes($password);  
 $password = htmlspecialchars($password);  
 //удаляем лишние пробелы  
 $login = trim($login);  
 $password = trim($password);  
 // подключаемся к базе  
 **include**("bd.php");// файл bd.php должен быть в той же папке, что и все остальные, если это не так, то просто измените путь  
  
 $result = mysqli\_query($db, "SELECT \* FROM users WHERE login='$login'"); //извлекаем из базы все данные о пользователе с введенным логином  
 $myrow = mysqli\_fetch\_array($result);  
 **if** (**empty**($myrow['password'])) {  
 //если пользователя с введенным логином не существует  
 **exit** ("<p style='color: red; text-align: center'>Введённый логин или пароль неверны</p>");  
 } **else** {  
 //если существует, то сверяем пароли  
 $password = hash('sha256', $password);  
 **if** ($myrow['password'] == $password) {  
 header("Location: main.php");  
 //если пароли совпадают, то запускаем пользователю сессию  
 $\_SESSION['login'] = $myrow['login'];  
 $\_SESSION['id'] = $myrow['id'];//эти данные очень часто используются, вот их и будет "носить с собой" вошедший пользователь  
  
 } **else** {  
 //если пароли не сошлись  
  
 **exit** ("<p style='color: red; text-align: center'>Введённый логин или пароль неверны</p>");  
 }  
 }  
 }  
 **?>** </div>  
</div>  
<footer>  
 <div class="footer\_vertical">  
 <p>Все права защищены</p>  
 <p>2018</p>  
 </div>  
</footer>  
  
</body>

bd.php

**<?php** $db = mysqli\_connect ("localhost","root","","tours");  
 **?>**

Main.php

**<?php**session\_start();  
**if** (**empty**($\_SESSION['login']) **or** $\_SESSION['login'] == '')  
{  
 header("Location: index.php");  
}  
**?>**<!DOCTYPE html>  
<html lang="en">  
  
<head>  
 <meta charset="UTF-8">  
 <title>Улётный Тур</title>  
 <link rel="shortcut icon" href="images/favicon.ico" type="image/x-icon">  
 <link rel="stylesheet" href="style.css">  
</head>  
  
<body>  
  
<div class="navbar">  
 <div class="header\_items header\_item\_1">  
 <a href="main.php">Главная</a>  
 </div>  
 <div class="header\_items header\_item\_2">  
 <a href="catalog.php">Каталог</a>  
 </div>  
 <div class="header\_items header\_item\_3">  
 <a href="cabinet.php">Личный кабинет</a>  
 </div>  
 <div class="header\_items header\_item\_4">  
 <a href="contacts.php">Контакты</a>  
 </div>  
 **<?php  
 if** (!**empty**($\_SESSION['login'])) {  
 **echo** ("<form class='header\_items' method='post' action='exit.php'>  
<input type='submit' name='exit' value='Выйти'>  
</form>");  
 }  
 **?>**</div>  
  
<header>  
 <div class="logo">  
 <img alt="Улётный Тур" src="images/logotype.png">  
 </div>  
 <div class="cabinet\_access">  
  
 </div>  
 <div class="contacts">  
 <p style="text-align: center;">Москва, ул. Туристическая, д.1, офис</p>  
 <p style="font-size:22px; font-weight: bold; text-align: center;">8 (800) 555-35-35</p>  
 </div>  
</header>  
  
<div class="motto">  
 <p>Найди свой тур прямо сейчас</p>  
</div>  
  
<h1 style="text-align: center">Горящие туры</h1>  
<div class="hot\_tours">  
 **<?php  
 include**("bd.php");  
 $result = mysqli\_query($db, "SELECT \* FROM tours\_list ORDER BY price ASC");  
 $rows = mysqli\_num\_rows($result);  
 **if** ($rows >= 4) $rows = 4;  
 **for** ($i = 0; $i < $rows; ++$i) {  
 $row = mysqli\_fetch\_array($result);  
 **echo** "<div class='hot\_tour\_item'><table>";  
 **echo** "<tr><th>$row[country]</th></tr>";  
 **echo** "<tr><td>Цена $row[price]</td></tr>";  
 **echo** "</table></div>";  
 }  
 **echo** "</div>";  
 **?>**</div>  
<footer>  
 <div class="footer\_vertical">  
 <p>Все права защищены</p>  
 <p>2018</p>  
 </div>  
</footer>  
</body>  
  
</html>

Catalog.php

**<?php**session\_start();  
**if** (**empty**($\_SESSION['login']) **or** $\_SESSION['login'] == '')  
{  
 header("Location: index.php");  
}  
**?>**<!DOCTYPE html>  
<html lang="ru">  
  
<head>  
 <meta charset="UTF-8">  
 <title>Улётный Тур - Каталог</title>  
 <link rel="shortcut icon" href="images/favicon.ico" type="image/x-icon">  
 <link rel="stylesheet" href="style.css">  
</head>  
  
<body>  
  
<div class="navbar">  
 <div class="header\_items header\_item\_1">  
 <a href="main.php">Главная</a>  
 </div>  
 <div class="header\_items header\_item\_2">  
 <a href="catalog.php">Каталог</a>  
 </div>  
 <div class="header\_items header\_item\_3">  
 <a href="cabinet.php">Личный кабинет</a>  
 </div>  
 <div class="header\_items header\_item\_4">  
 <a href="contacts.php">Контакты</a>  
 </div>  
 **<?php  
 if** (!**empty**($\_SESSION['login'])) {  
 **echo** ("<form class='header\_items' method='post' action='exit.php'>  
<input type='submit' name='exit' value='Выйти'>  
</form>");  
 }  
 **?>**</div>  
  
<header>  
 <div class="logo">  
 <img alt="Улётный Тур" src="images/logotype.png">  
 </div>  
 <div class="cabinet\_access">  
  
 </div>  
 <div class="contacts">  
 <p style="text-align: center;">Москва, ул. Туристическая, д.1, офис</p>  
 <p style="font-size:22px; font-weight: bold; text-align: center;">8 (800) 555-35-35</p>  
 </div>  
</header>  
**<?php  
if** ($\_SESSION['login'] == '123dobro@mail.ru' **or** $\_SESSION['login'] == '123zlodey@mail.ru') {  
 **echo**('<form class="cat\_form" method="post">  
<table>  
<tr><td></td><th style="text-align: center"><p>Добавить новый тур</p></th></tr>  
 <tr>  
 <td>  
 <input class="catalog\_input" name="tour\_name" placeholder="Название">  
 </td>  
 <td>  
 <input class="catalog\_input" name="tour\_description" placeholder="Описание">  
 </td>  
 <td>  
 <input class="catalog\_input" name="tour\_price" pattern="^[ 0-9]+$" placeholder="Стоимость">  
 </td>  
 </tr>  
   
 <tr><td></td>  
 <td class="btn\_row">  
 <input type="submit" class="log\_button" name="add\_tour" value="Добавить">  
 </td>  
</tr>  
   
 <tr><td></td><th style="text-align: center"><p>Удалить тур</p></th></tr>  
 <tr>  
 <td>  
 <input class="catalog\_input" name="tour\_id" pattern="^[ 0-9]+$" placeholder="Номер">  
 </td>  
 <td>  
 <input class="catalog\_input" name="tour\_name\_del" placeholder="Название">  
 </td>  
  
</tr>  
 <tr><td></td>  
 <td class="btn\_row">  
 <input type="submit" class="log\_button" name="del\_tour" value="Удалить">  
 </td>  
</tr>  
  
<tr><td></td><th style="text-align: center"><p>Редактировать тур</p></th></tr>  
<tr>  
 <td>  
 <input class="catalog\_input" name="tour\_id\_red" pattern="^[ 0-9]+$" placeholder="Номер">  
 </td>  
 <td>  
 <input class="catalog\_input" name="tour\_name\_red" placeholder="Название">  
 </td>  
</tr>  
 <tr><td></td>  
 <td class="btn\_row">  
 <input type="submit" class="log\_button" name="red\_tour" value="Изменить">  
 </td>  
</tr>  
</table>  
</form>');  
}  
**?>  
  
<?php  
if** (**isset**($\_POST['add\_tour'])) {  
 **if** (**isset**($\_POST['tour\_name'])) {  
 $tour\_name = $\_POST['tour\_name'];  
 $tour\_name = stripslashes($tour\_name);  
 $tour\_name = htmlspecialchars($tour\_name);  
 $tour\_name = trim($tour\_name);  
 **if** ($tour\_name == '') {  
 **unset**($tour\_name);  
 }  
 } //заносим введенный пользователем логин в переменную $login, если он пустой, то уничтожаем переменную  
 **if** (**isset**($\_POST['tour\_description'])) {  
 $tour\_description = $\_POST['tour\_description'];  
 $tour\_description = stripslashes($tour\_description);  
 $tour\_description = htmlspecialchars($tour\_description);  
 $tour\_description = trim($tour\_description);  
 **if** ($tour\_description == '') {  
 **unset**($tour\_description);  
 }  
 }  
 **if** (**isset**($\_POST['tour\_price'])) {  
 $tour\_price = $\_POST['tour\_price'];  
 $tour\_price = stripslashes($tour\_price);  
 $tour\_price = htmlspecialchars($tour\_price);  
 $tour\_price = trim($tour\_price);  
 **if** ($tour\_price == '') {  
 **unset**($tour\_price);  
 }  
 }  
 **if** (**empty**($tour\_name) **or empty**($tour\_description) **or empty**($tour\_price)) {  
 **exit** ("<p style='color: red; text-align: center'>Заполните все поля!</p>");  
 }  
 **include**("bd.php");// файл bd.php должен быть в той же папке, что и все остальные, если это не так, то просто измените путь  
  
 $result = mysqli\_query($db, "SELECT \* FROM tours\_list WHERE country='$tour\_name'");  
 $myrow = mysqli\_fetch\_array($result);  
 **if** ($myrow['description'] == $tour\_description) {  
 **exit** ("<p style='color: red; text-align: center'>Такой тур уже существует</p>");  
 } **else** {  
 $result2 = mysqli\_query($db, "INSERT INTO tours\_list (country,description,price) VALUES('$tour\_name','$tour\_description','$tour\_price')");  
 }  
}  
**?>  
  
<?php  
if** (**isset**($\_POST['del\_tour'])) {  
 **if** (**isset**($\_POST['tour\_name\_del'])) {  
 $tour\_name\_del = $\_POST['tour\_name\_del'];  
 **if** ($tour\_name\_del == '') {  
 **unset**($tour\_name\_del);  
 }  
 } //заносим введенный пользователем логин в переменную $login, если он пустой, то уничтожаем переменную  
 **if** (**isset**($\_POST['tour\_id'])) {  
 $tour\_id = $\_POST['tour\_id'];  
 **if** ($tour\_id == '') {  
 **unset**($tour\_id);  
 }  
 }  
 **if** (**empty**($tour\_id) **or empty**($tour\_name\_del)) {  
 **exit** ("<p style='color: red; text-align: center'>Заполните все поля!</p>");  
 }  
 $tour\_name\_del = stripslashes($tour\_name\_del);  
 $tour\_name\_del = htmlspecialchars($tour\_name\_del);  
 $tour\_id = stripslashes($tour\_id);  
 $tour\_id = htmlspecialchars($tour\_id);  
 $tour\_name\_del = trim($tour\_name\_del);  
 $tour\_id = trim($tour\_id);  
  
 **include**("bd.php");  
 $result = mysqli\_query($db, "SELECT \* FROM tours\_list WHERE id='$tour\_id'");  
 $myrow = mysqli\_fetch\_array($result);  
 **if** (**empty**($myrow)) {  
 **exit** ("<p style='color: red; text-align: center'>Такого тура не существует</p>");  
 } **else** {  
 $result2 = mysqli\_query($db, "DELETE FROM tours\_list WHERE id = '$tour\_id'");  
 }  
}  
**?>  
  
<?php  
if** (**isset**($\_POST['red\_tour'])) {  
 **if** (**isset**($\_POST['tour\_name\_red'])) {  
 $tour\_name\_red = $\_POST['tour\_name\_red'];  
 **if** ($tour\_name\_red == '') {  
 **unset**($tour\_name\_red);  
 }  
 } //заносим введенный пользователем логин в переменную $login, если он пустой, то уничтожаем переменную  
 **if** (**isset**($\_POST['tour\_id\_red'])) {  
 $tour\_id\_red = $\_POST['tour\_id\_red'];  
 **if** ($tour\_id\_red == '') {  
 **unset**($tour\_id\_red);  
 }  
 }  
 **if** (**empty**($tour\_id\_red) **or empty**($tour\_name\_red)) {  
 **exit** ("<p style='color: red; text-align: center'>Заполните все поля!</p>");  
 }  
 $tour\_name\_red = stripslashes($tour\_name\_red);  
 $tour\_name\_red = htmlspecialchars($tour\_name\_red);  
 $tour\_id\_red = stripslashes($tour\_id\_red);  
 $tour\_id\_red = htmlspecialchars($tour\_id\_red);  
 $tour\_name\_red = trim($tour\_name\_red);  
 $tour\_id\_red = trim($tour\_id\_red);  
  
 **include**("bd.php");  
 $result = mysqli\_query($db, "SELECT \* FROM tours\_list WHERE country='$tour\_name\_red'");  
 $myrow = mysqli\_fetch\_array($result);  
 **if** ($myrow['id'] != $tour\_id\_red) {  
 **exit** ("<p style='color: red; text-align: center'>Такого тура не существует</p>");  
 } **else** {  
 **echo**("<form class='cat\_form' method='post'>  
<table>  
 <tr>  
 <td>  
 <input class='catalog\_input' name='tour\_id\_red' value='$tour\_id\_red' readonly>  
</td>  
 <td>  
 <input class='catalog\_input' name='tour\_re\_name' placeholder='Название'>  
 </td>  
 <td>  
 <input class='catalog\_input' name='tour\_re\_description' placeholder='Описание'>  
 </td>  
 <td>  
 <input class='catalog\_input' name='tour\_re\_price' pattern=\"^[ 0-9]+$\" placeholder='Цена'>  
 </td>  
   
 <td class='btn\_row'>  
 <input type='submit' class='log\_button' name='re\_add\_tour' value='Сохранить'>  
 </td>  
 </tr>  
  
</table>  
</form>");  
 }  
}  
**?>  
  
<?php  
if** (**isset**($\_POST['re\_add\_tour'])) {  
  
 $tour\_id\_red = $\_POST['tour\_id\_red'];  
  
 **if** (**isset**($\_POST['tour\_re\_name'])) {  
 $tour\_re\_name = $\_POST['tour\_re\_name'];  
 $tour\_re\_name = stripslashes($tour\_re\_name);  
 $tour\_re\_name = htmlspecialchars($tour\_re\_name);  
 $tour\_re\_name = trim($tour\_re\_name);  
  
 } //заносим введенный пользователем логин в переменную $login, если он пустой, то уничтожаем переменную  
 **if** (**isset**($\_POST['tour\_re\_description'])) {  
 $tour\_re\_description = $\_POST['tour\_re\_description'];  
 $tour\_re\_description = stripslashes($tour\_re\_description);  
 $tour\_re\_description = htmlspecialchars($tour\_re\_description);  
 $tour\_re\_description = trim($tour\_re\_description);  
 }  
 **if** (**isset**($\_POST['tour\_re\_price'])) {  
 $tour\_re\_price = $\_POST['tour\_re\_price'];  
 $tour\_re\_price = stripslashes($tour\_re\_price);  
 $tour\_re\_price = htmlspecialchars($tour\_re\_price);  
 $tour\_re\_price = trim($tour\_re\_price);  
 }  
  
 **include**("bd.php");  
 **if** ($tour\_re\_name != '') {  
 $result = mysqli\_query($db, "UPDATE tours\_list SET country='$tour\_re\_name' WHERE id='$tour\_id\_red'");  
 }  
 **if** ($tour\_re\_description != '') {  
 $result = mysqli\_query($db, "UPDATE tours\_list SET description='$tour\_re\_description' WHERE id='$tour\_id\_red'");  
 }  
 **if** ($tour\_re\_price != '') {  
 $result = mysqli\_query($db, "UPDATE tours\_list SET price='$tour\_re\_price' WHERE id='$tour\_id\_red'");  
 }  
}  
**?>**<div class="catalog\_main">  
 **<?php  
 include**("bd.php");  
 $result = mysqli\_query($db, "SELECT \* FROM tours\_list");  
 $rows = mysqli\_num\_rows($result); // количество полученных строк  
 **echo** "<table style='margin-top: 30px' cellspacing='0'><tr><th class='t\_header'>Номер</th>  
 <th class='t\_header'>Название</th>  
 <th class='t\_header'>Описание</th>  
 <th class='t\_header'>Цена</th>  
 <th class='t\_header'></th></tr>";  
 **for** ($i = 0; $i < $rows; ++$i) {  
 $myrow = mysqli\_fetch\_array($result);  
 **echo** "<form method='post' action='cabinet.php'>";  
 **echo** "<tr class='border\_bottom'>";  
 **echo**("<td><input class='catalog\_text' name='order\_id' value='$myrow[0]' readonly style='width: 50px'></td>");  
 **echo**("<td><input class='catalog\_text' value='$myrow[1]' readonly style='font-weight: bold'></td>");  
 **echo**("<td><p class='catalog\_text\_area'>$myrow[2]</p></td>");  
 **echo**("<td><input class='catalog\_text' value='$myrow[3]' readonly></td>");  
 **echo** "<td><input type='submit' class='buy\_button' name='buy\_button' value='Купить'></td>";  
 **echo** "</tr>";  
 **echo** "</form>";  
 }  
 **echo** "</table>";  
 **?>**</div>  
<footer>  
 <div class="footer\_vertical">  
 <p>Все права защищены</p>  
 <p>2018</p>  
 </div>  
</footer>  
</body>

Cabinet.php

**<?php**session\_start();  
**if** (**empty**($\_SESSION['login']) **or** $\_SESSION['login'] == '')  
{  
 header("Location: index.php");  
}  
**?>**<!DOCTYPE html>  
<html lang="en">  
  
<head>  
 <meta charset="UTF-8">  
 <title>Улётный Тур - Личный кабинет</title>  
 <link rel="shortcut icon" href="images/favicon.ico" type="image/x-icon">  
 <link rel="stylesheet" href="style.css">  
</head>  
  
<body>  
  
<div class="navbar">  
 <div class="header\_items header\_item\_1">  
 <a href="main.php">Главная</a>  
 </div>  
 <div class="header\_items header\_item\_2">  
 <a href="catalog.php">Каталог</a>  
 </div>  
 <div class="header\_items header\_item\_3">  
 <a href="cabinet.php">Личный кабинет</a>  
 </div>  
 <div class="header\_items header\_item\_4">  
 <a href="contacts.php">Контакты</a>  
 </div>  
 **<?php  
 if** (!**empty**($\_SESSION['login'])) {  
 **echo** ("<form class='header\_items' method='post' action='exit.php'>  
 <input type='submit' name='exit' value='Выйти'>  
 </form>");  
 }  
 **?>**</div>  
  
<header>  
 <div class="logo">  
 <img alt="Улётный Тур" src="images/logotype.png">  
 </div>  
 <div class="cabinet\_access">  
  
 </div>  
 <div class="contacts">  
 <p style="text-align: center;">Москва, ул. Туристическая, д.1, офис</p>  
 <p style="font-size:22px; font-weight: bold; text-align: center;">8 (800) 555-35-35</p>  
 </div>  
</header>  
  
  
<div class="orders">  
  
 **<?php  
 if** ($\_SESSION['login'] == '123dobro@mail.ru') {  
 **echo**('<div style="display: flex; flex-direction: column"><h1 style="text-align: center">Отчёт по заказам</h1><div style="display: flex; flex-direction: column; align-items: center;"  
 <table>  
 <form class="cat\_form" method="post" action="del.php">  
 <tr>  
 <td>  
 <input style="margin-right: 10px" class="catalog\_input" name="user\_id" pattern="^[ 0-9]+$" placeholder="id">  
 </td>  
 <td>  
 <input style="margin-right: 10px" class="catalog\_input" name="user\_name" placeholder="login">  
 </td>  
 <td><input style="padding: 18px; margin: 0" type="submit" class="log\_button btn\_row" name="submit" value="Удалить"></td>  
 </tr>  
   
 </form>  
 </table>  
 ');  
 **include**("bd.php");  
 $result = mysqli\_query($db, "SELECT \* FROM users");  
 $rows = mysqli\_num\_rows($result); // количество полученных строк  
 **echo** "<table><tr><th class='t\_header'>id</th><th class='t\_header'>login</th><th class='t\_header'>Номер заказа</th></tr>";  
 **for** ($i = 0; $i < $rows; ++$i) {  
 $row = mysqli\_fetch\_array($result);  
 **echo** "<tr>";  
 **echo** "<td>$row[0]</td>";  
 **echo** "<td>$row[1]</td>";  
 **if** ($row[3] == 0) $row = **null**;  
 **echo** "<td>$row[3]</td>";  
 **echo** "</tr>";  
 }  
 **echo** "</div></div>";  
 }  
 **?>  
  
  
 <?php  
 if** ($\_SESSION['login'] != '123dobro@mail.ru') {  
 **include**("bd.php");  
 **if** (!**empty**($\_POST['order\_id'])) {  
 $result = mysqli\_query($db, "SELECT \* FROM tours\_list WHERE id = '$\_POST[order\_id]'");  
 $rows = mysqli\_num\_rows($result); // количество полученных строк  
 } **else** {  
 $get\_order\_id = mysqli\_query($db, "SELECT \* FROM users WHERE id = '$\_SESSION[id]'");  
 $get\_order\_id = mysqli\_fetch\_array($get\_order\_id);  
 $\_SESSION['paid'] = $get\_order\_id['paid'];  
 $result = mysqli\_query($db, "SELECT \* FROM tours\_list WHERE id = '$get\_order\_id[orders]'");  
 $rows = mysqli\_num\_rows($result);  
 $result = mysqli\_fetch\_array($result); // количество полученных строк  
 }  
 **if** (**empty**($rows) **or** $rows == 0) **exit** ("<p style='font-weight: bold; font-size: 20px; color: grey; margin-top: 30px'>Заказов нет</p>");  
 **echo** "<table style='margin-top: 30px' cellspacing='0'><tr>  
 <th class='t\_header'>Номер</th>  
 <th class='t\_header'>Название</th>  
 <th class='t\_header'>Описание</th>  
 <th class='t\_header'>Дата</th>  
 <th class='t\_header'>Оплатить</th>  
 <th class='t\_header'>Отменить</th></tr>";  
 **for** ($i = 0; $i < $rows; ++$i) {  
 **echo** "<form method='post' action='buy.php'>";  
 **echo** "<tr class='border\_bottom'>";  
 **echo**("<td><input class='catalog\_text' name='order\_id' value='$result[0]' readonly style='width: 50px'></td>");  
 **echo**("<td><input class='catalog\_text' value='$result[1]' readonly style='font-weight: bold'></td>");  
 **echo**("<td><p class='catalog\_text\_area'>$result[2]</p></td>");  
 **if** (**empty**($\_SESSION['paid'])) {  
 **echo**("<td><input type='date' class='catalog\_text' name='calendar' value='" . date("Y-m-d") . "' min='" . date("Y-m-d") . "'></td>");  
 **echo** "<td class='catalog\_text'><input type='submit' class='buy\_button' name='buy\_button' value='Оплатить'></td>";  
 **echo** "<td class='catalog\_text'><input type='submit' class='refuse\_button' name='refuse\_button' value='Отказаться'></td>";  
 } **else** {  
 **echo** "<td class='catalog\_text'><p style='padding: 0 20px;'>$\_SESSION[date]</p></td>";  
 **echo** "<td class='catalog\_text'><p style='padding: 0 5px;'>Оплачено</p></td>";  
 **echo** "<td class='catalog\_text'><input type='submit' class='refuse\_button' name='refuse\_button' value='Отказаться'></td>";  
 }  
 **echo** "</tr>";  
 **echo** "</form>";  
 }  
 **echo** "</table>";  
 }  
 **?>**</div>  
  
</body>

Del.php

**<?php**session\_start();  
**if** (**empty**($\_SESSION['login']) **or** $\_SESSION['login'] == '')  
{  
 header("Location: index.php");  
}  
**include**("bd.php");  
$result = mysqli\_query($db, "SELECT \* FROM users WHERE id='$\_POST[user\_id]'");  
$myrow = mysqli\_fetch\_array($result);  
**if** (**empty**($myrow) **or** $myrow['login']!=$\_POST['user\_name']) {  
 header("Location: cabinet.php");  
} **else** {  
 $result2 = mysqli\_query($db, "DELETE FROM users WHERE id = '$\_POST[user\_id]'");  
 header("Location: cabinet.php");  
}

Buy.php

**<?php**session\_start();  
**if** (**empty**($\_SESSION['login']) **or** $\_SESSION['login'] == '')  
{  
 header("Location: index.php");  
}  
  
**if** (**isset**($\_POST['buy\_button'])) {  
 **include** ("bd.php");// файл bd.php должен быть в той же папке, что и все остальные, если это не так, то просто измените путь  
 $result = mysqli\_query($db, "UPDATE users SET orders = '$\_POST[order\_id]' WHERE id='$\_SESSION[id]'");  
 $result = mysqli\_query($db, "UPDATE users SET paid = '1' WHERE id='$\_SESSION[id]'");  
 $\_SESSION['paid'] = **true**;  
 $\_SESSION['date'] = $\_POST['calendar'];  
}  
**if** (**isset**($\_POST['refuse\_button'])) {  
 **unset**($\_POST);  
 **include** ("bd.php");// файл bd.php должен быть в той же папке, что и все остальные, если это не так, то просто измените путь  
 $result = mysqli\_query($db, "UPDATE users SET orders = '' WHERE id='$\_SESSION[id]'");  
}  
header("Location: cabinet.php");  
**?>**

# **ЗАКЛЮЧЕНИЕ**

В результате выполнения курсовой работы по дисциплине «Разработка программных приложений» на тему «Информационно-справочная система "Автомобили автосалона“» было выполнены следующие действия:

1. Разработка программного продукта.
2. Разработка и подробное описание структуры программного продукта.
3. Проведено тестирование (проверка) программы на локальном сервере.
4. Произведено размещение справочной системы на хостинге в сети Интернет.
5. Написана пояснительная записка к курсовому проекту.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

* 1. Лабораторные и практические материалы по курсу
  2. https://geekbrains.ru/chapters/927
  3. Справочные материалы http://htmlbook.ru/ и http://php.net