

**МИНИСТЕРСТВО СВЯЗИ, ИНФОРМАТИЗАЦИИ И  
ТЕЛЕКОМУНИКАЦИОННЫХ ТЕХНОЛОГИЙ РЕСПУБЛИКИ УЗБЕКИСТАН  
ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

К защите допустить  
Зав. кафедрой

\_\_\_\_\_ 2015 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**на тему:**

**Разработка веб сайта авто лизинговой компании**

Выпускник \_\_\_\_\_ Тухтасинов А.О.  
подпись Ф.И.О.

Руководитель \_\_\_\_\_ Керимов К.Ф.  
подпись Ф.И.О.

Консультант по БЖД \_\_\_\_\_ Абдуллаева С.М.  
подпись Ф.И.О.

Рецензент \_\_\_\_\_ ..  
подпись Ф.И.О.



**МИНИСТЕРСТВО СВЯЗИ, ИНФОРМАТИЗАЦИИ И  
ТЕЛЕКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ РУЗ  
ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ**

Факультет: Программный инжиниринг  
Кафедра: Системное и прикладное программирование  
Направление (специальность):  
5521900 – «Информатика и информационные технологии»

**У Т В Е Р Ж Д А Ю**

Зав. кафедрой \_\_\_\_\_  
« \_\_\_\_ » \_\_\_\_\_ 2015г.

**ЗАДАНИЕ**

на выпускную квалификационную работу студента Тухтасинова  
Авазбека Ойбек угли

(фамилия, имя, отчество)

1. Тема работы: Разработка веб сайта авто лизинговой компании.

2. Тема утверждена приказом по университету от « 00 » 00 0000 г. № 000-00

3. Срок сдачи законченной работы: 0.00.0000.

4. Исходные данные к работе:

Задания по постановке задачи, технические данные, данные по продукции  
на лизинг, литература, учебные пособия, диаграммы, интернет-  
ресурсы

5. Содержание расчётно-пояснительной записки (перечень подлежащих к  
разработке вопросов): Анализ предметной области, анализ требований к  
разработке веб сайта, проектирование структуры БД, проектирование  
архитектуры веб сайта, реализация веб сайта на языке программирования.

6. Перечень графического материала: таблицы, пользовательские интерфейсы  
(скриншоты), диаграммы, рисунки, презентация.

7. Дата выдачи задания \_\_\_\_\_

Руководитель \_\_\_\_\_  
подпись

Задание принял \_\_\_\_\_  
подпись

## 8. Консультанты по отдельным разделам выпускной работы

Раздел	Ф.И.О руководителя	Подпись, дата	
		Задание выдал	Задание получил
Основная часть	Керимов К. Ф.		
БЖД	Абдуллаева С.М.		

## 9. График выполнения работы

Наименование раздела работы	Срок выполнения	Отметка руководителя о выполнении
1. Введение	27.02.2014 г – 28.02.2014 г	
2. Анализ существующих веб сайтов	1.03.2014 г – 12.03.2014 г	
3. Формирование постановки задачи	13.03.2014 г – 15.03.2014 г	
4. Реализация базы данных	16.03.2014 г – 20.03.2014 г	
5. Реализация веб сайта	21.03.2014 г – 20.04.2014 г	
6. Описание веб сайта	21.04.2014 г – 29.04.2014 г	
7. Подготовка презентации	30.04.2014 г – 10.05.2014 г	
8. Написание доклада	11.05.2014 г – 25.05.2014 г	

Выпускник \_\_\_\_\_ « \_\_\_\_\_ » \_\_\_\_\_ 2015 г.  
(подпись)

Руководитель \_\_\_\_\_ « \_\_\_\_\_ » \_\_\_\_\_ 2015 г.  
(подпись)

## **Аннотация**

Данная выпускная квалификационная работа посвящена разработке веб сайта авто лизинговой компании. В данной работе были рассмотрены теоретические основы построения веб сайтов и методы реализации таких сайтов, определены проблемы и выработаны требования для реализации, приведена постановка задачи и практическая реализация системы. Была разработана и заполнена база данных. При разработке были использованы язык программирования PHP, в теле системы управления контентом Drupal, язык разметки HTML, каскадные таблицы стилей CSS, СУБД MySQL, а также технологии FlexSlider, Fancybox, JQuery и Bootstrap.

## **Annotation**

This Final qualification work is devoted to develop the web site of auto leasing company. In this qualification work the theoretical foundations of web site foundation and implementation methods for these systems are described, the problems were identified and the requirements for implementation were worked out, the problems and the practical implementation of the system are given. The database was developed and filled. Programming language PHP, in body of CMS Drupal, markup language HTML, cascading style sheets CSS, DBMS MySQL, and also FlexSlider, Fancybox, JQuery and Bootstrap were used in developing the web site.

## **Мазмуннома**

Mazkur bitiruv malakaviy ishi avto lizing kompaniyasi veb saytiga bag'ishlangan. Ushbu ishda veb sayt qurish asoslari va amalga oshirish usullari, muammolarni aniqlab ularni xal qilish yechimlarini talablari va amaliy bajarilishini ko'rib chiqildi. Yaratish uchun Drupal – kontentni boshqarish tizimi negizida PHP dasturlash tili, HTML, CSS, MySQL MBBT, FlexSlider, Fancybox, JQuery, va Bootstrap texnologiyalardan foydalanildi.

## ОГЛАВЛЕНИЕ

<b>ВВЕДЕНИЕ .....</b>	<b>7</b>
<b>1. Анализ существующих веб сайтов .....</b>	<b>9</b>
1.1. Теоретические основы.....	9
1.2. Структура любого сайта.....	13
1.3. Анализ технологий и методов построения веб сайтов .....	20
<b>2. Проектирование системы.....</b>	<b>22</b>
2.1. Требования к программе .....	22
2.2. Стадии и этапы разработки.....	223
2.3. Постановка задачи .....	24
2.4. Структурная схема базы данных.....	37
<b>3. Описание веб-сайта .....</b>	<b>40</b>
3.1. Архитектура системы .....	40
3.2. Навигация .....	41
3.4. Онлайн заявка.....	42
3.5. Процесс добавления новых данных .....	4463
3.6. Просмотр предложений.....	46
<b>4. Безопасность жизнедеятельности.....</b>	<b>48</b>
4.1. Взаимодействие человека и техносферы .....	48
4.2. Пожарная безопасность.....	53
4.2. Техногенное загрязнение среды.....	61
<b>Заключение.....</b>	<b>48</b>
<b>Список использованной литературы.....</b>	<b>49</b>

## ВВЕДЕНИЕ

Президент Республики Узбекистан И.А. Каримов: XXI век – век информационно-коммуникационных технологий и интернета.

XXI век становится веком глобализации и стирания границ, информационно-коммуникационных технологий и интернета, веком все более растущей конкуренции на мировом пространстве и мировом рынке.

Об этом на открытии международной конференции «Подготовка образованного и интеллектуально развитого поколения – как важнейшее условие устойчивого развития и модернизации страны», состоявшейся в феврале 2012 года, заявил Президент Республики Узбекистан Ислам Каримов.

«В настоящее время все образовательные, научные и культурно-просветительские учреждения страны подключены к образовательному порталу и интернету, – подчеркнул глава государства. – Это позволяет внедрять в образовательный процесс методы дистанционного обучения, широкий комплекс информационно-коммуникационных услуг для учащейся молодежи»[1].

Для бизнесменов, которые столь важны для процветания нашего государства, как основной фундамент налогоплательщиков, должны создаваться все условия и удобства, чтобы их бизнес развивался. Покупка автомобиля в лизинг облегчит нелегкий путь ведения бизнеса, а создание сайта для автолизинговой компании – упростит задачу поиска и сравнения вариаций для нужной продукции.

Веб сайт автолизинговой компании предназначен для того, чтобы люди с легкостью находили нужный им автомобиль и решить проблему длинного периода выбора и договора, связанных с покупкой автомобиля в лизинг.

Главной целью данного сайта является удовлетворение потребностей предприятий, у которых стоит проблема покупки автомобиля в лизинг.

## Назначение разработки

Целью настоящей ВКР является разработка веб сайта для авто лизинговой компании.

*Программа предназначена для:*

- Сбора информации об автомобилях, которые продаются в лизинг, и создания базы автомобилей;
- использования методов сайта для предоставления полной информации об автомобилях потенциальным покупателям;
- ускорения процесса выбора автомобилей и связи с компанией-продавцом;
- поиска автомобилей по базе сайта;

*Во время разработки будет проведено:*

- изучение и оценка использования различных сайтов;
- разработка структуры входных и выходных данных;

Задачи ВКР:

- анализ данной предметной области
- создание системы вывода данных и связи пользователей с администратором
- сопровождение и поддержка данной веб сайта

# 1. Анализ существующих сайтов

## 1.1. Теоретические основы

**Сайт** (от англ. *website*: *web* — «паутина, сеть» и *site* — «место», буквально «место, сегмент, часть в сети») — система электронных документов (файлов данных и кода) частного лица или организации в компьютерной сети под общим адресом (доменным именем или IP-адресом)

Все сайты в совокупности составляют Всемирную паутину, где коммуникация (паутина) объединяет сегменты информации мирового сообщества в единое целое — базу данных и коммуникации планетарного масштаба. Для прямого доступа клиентов к сайтам на серверах был специально разработан протокол HTTP.

Первый в мире сайт, [info.cern.ch](http://info.cern.ch), появился в 1991 году. Его создатель, Тим Бернерс-Ли, опубликовал на нём описание новой технологии World Wide Web, основанной на протоколе передачи данных HTTP, системе адресации URI и языке гипертекстовой разметки HTML. Также на сайте были описаны принципы установки и работы серверов и браузеров. Сайт стал, и первым в мире интернет-каталогом, так как позже Тим Бернерс-Ли разместил на нём список ссылок на другие сайты.

Страницы сайтов — это набор текстовых файлов, размеченных на языке HTML. Эти файлы, будучи загруженными посетителем на его компьютер, понимаются и обрабатываются *браузером* и выводятся на средство отображения пользователя (монитор, экран КПК, принтер или синтезатор речи). Язык HTML позволяет форматировать текст, различать в нём функциональные элементы, создавать гипертекстовые ссылки (*гиперссылки*) и вставлять в отображаемую страницу изображения, звукозаписи и другие мультимедийные элементы. Отображение страницы можно изменить добавлением стилей на языке CSS, что

позволяет централизовать в определенном файле все элементы форматирования (размер и цвет заглавных букв 2-го уровня, размер и вид блока вставки и другое) или сценариев на языке JavaScript, с помощью которого имеется возможность просматривать страницы с событиями или действиями.

Определённый класс сайтов иначе называют интернет-представительством человека или организации. Как комментарий к ссылке может быть страничка-визитка на полнофункциональном сайте (портале). Когда говорят «своя страничка в Интернете», то подразумевают целый сайт или личную страницу в составе чужого сайта (портал). Кроме сайтов (порталов), в сети Интернет также доступны WAP-сайты для мобильных телефонов.

В большинстве случаев в Интернете одному сайту соответствует одно доменное имя. Именно по доменным именам сайты идентифицируются в глобальной сети. Возможны иные варианты: один сайт на нескольких доменах или несколько сайтов под одним доменом. Обычно несколько доменов используют крупные сайты (веб-порталы), чтобы логически отделить разные виды предоставляемых услуг (mail.google.com, news.google.com, maps.google.com). Нередки и случаи выделения отдельных доменов для разных стран или языков. Например, google.ru и google.fr логически являются сайтом Google на разных языках, но технически это разные сайты. Объединение нескольких сайтов под одним доменом характерно для бесплатных хостингов. Иногда для идентификации сайтов в адресе после указания хоста стоит тильда и имя сайта: example.com/~my-site-name/ (ср. с /home), а чаще всего используется домен третьего уровня: my-site-name.example.com.

Аппаратные серверы для хранения сайтов называются *веб-серверами*. Сама услуга хранения называется *хостингом*. Раньше каждый сайт хранился на своём собственном сервере, но с ростом Интернета, технологическим улучшением

серверов на одном компьютере стало возможно размещение множества сайтов (виртуальный хостинг). Сейчас серверы для хранения только одного сайта называются *выделенными* (англ. *dedicated*).

**Сайт-визитка** (англ. *business card website*) — небольшой сайт, как правило, состоящий из одной (или нескольких) веб-страницы, и содержащий основную информацию об организации, частном лице, компании, товарах или услугах, прайс-листы, контактные данные.

Часто сайт-визитку используют компании, которые не хотят нести большие затраты на создание отдельного сайта. Очень часто при покупке доменного имени для почты устанавливается сайт-заглушка или сайт-визитка.

## 1.2. Структура любого сайта

Процесс создания любого сайта или его реконструкции начинается с разработки структуры. Необходимо сформулировать сколько и каких разделов и подразделов целесообразно сделать, сколько страниц должно быть в каждом из разделов (подразделов). Особое внимание целесообразно уделить полноте охвата, логике подачи материала.

Такая классификация, конечно, является общей, и структура многих сайтов может представлять собой комбинацию основных видов, а иногда и совсем не подходить ни под одно из описаний.

### **Линейная структура**

Это самая элементарная структура сайта. Веб-страницы идут одна за другой и пользователь должен просматривать их как слайд-шоу. В линейной структуре не существует деления контента (страниц) на уровни. Все страницы на таких сайтах равноправны, и их последовательно должен увидеть каждый посетитель.

Самая простая структура. Страницы идут одна за другой, пользователь может просматривать их, как слайд-шоу. Использование такого типа структур ограничено, их применяют в основном на имиджевых сайтах-презентациях и в онлайн-учебных пособиях, книгах. Реализовать такой вид структуры очень легко, так как в большинстве случаев она представляет собой набор html-страниц, с каждой из которых есть ссылка на следующую(предыдущую). Но даже при создании сайта с такой простой структурой веб-мастера часто допускают ошибки, которые ставят пользователя в тупик. Очень важно, чтобы на каждой странице такого сайта были название и ссылка на первую страницу, желательно также указывать общее количество страниц и обозначать ту, на которой в данный момент пользователь находится. Таким образом ,

создается своеобразная карта проекта, которая помогает посетителю осуществлять путешествие по сайту.

### *Область применения*

Область применения сайтов с линейной структурой четко ограничена. Она может использоваться на имиджевых сайтах, сайтах-презентациях, в онлайн-учебных пособиях и электронных книгах (e-book).

### **Линейная структура с альтернативными вариантами**

Этот вид структуры очень похож на линейную структуру, с тем лишь отличием, что пользователи имеют больше вариантов для поиска информации, а точнее - выбор между 2-мя ветками.

Основой данной структуры является простое линейное размещение web-страниц. Однако посетители могут проявить некоторую инициативу, облегчив для себя поиск нужной информации. Под альтернативами в данном случае понимается выбор между двумя ветками.

### *Область применения*

Чаще всего подобная структура используется для заполнения форм регистрации посетителей сайта. В этом случае все люди начинают работу со стартовой страницы. Однако потом частным лицам предлагается ввести одну информацию, а представителям коммерческих структур — другую. После этого и те, и другие попадают на одну и ту же страницу.

Линейная структура с альтернативами и вариантами удобна в том плане, что с одной стороны она позволяет веб-мастерам контролировать деятельность посетителей, направив их в определенное русло. А с другой стороны проявить

некоторую инициативу, которая позволит им, во-первых, почувствовать свободу, а во-вторых, облегчить доступ к нужной именно им информации. Например, переход на следующую страницу без заполнения текущей страницы.

### **Линейная структура с ответвлениями**

Данная структура аналогична дороге с множественными ответвлениями от неё. Человек переходит с одной страницы на другую в строгой последовательности как при линейной структуре. Однако посетитель в случае необходимости может всегда перейти на другое ответвление, а затем возвратиться назад.

Основным достоинством линейной структуры с ответвлениями – это возможность перейти на неё с обычной линейной структуры. В этом часто возникает необходимость, когда на сайте сильно разрастается контент и посетителям становится сложнее искать определенные статьи.

Это тоже контролируемая структура, которая напоминает дорогу с ответвляющимися от нее время от времени тупиковыми тропинками. То есть посетитель последовательно переходит с одной страницы на другую. Если информация, размещенная на какой-то из них его заинтересовала и он хочет узнать подробности, то может перейти на ответвление, а потом вернуться обратно на основную «дорожку».

Главным преимуществом рассматриваемой структуры является то, что к ней легко перейти с обычного линейного размещения веб-страниц. Такое часто бывает, когда созданный однажды сайт перестает удовлетворять возросшим требованиям, а глобальная переделка по тем или иным причинам невозможна. В этом случае web-мастер может быстро и без всяких проблем расширить проект.

В основе данного вида лежит простое последовательное размещение веб-страниц. Но в отличие от простой линейной структуры пользователю предоставляется не только возможность движения от первой к последней странице, а также и выбор другой ветки.

### **Решётчатая структура**

Одна из самых сложных структур сайта, где все документы располагаются в разных ветках. Однако посетитель может легко перемещаться по этим веткам как горизонтально (слева направо или между ветками на различных уровнях), так и вертикально (сверху вниз).

Данный вид структуры характерен преимущественно для каталогов статей или ссылок. На первый взгляд она очень удобна для пользователей, но для обычных сайтов её лучше не использовать.

Дело в том, что решётчатую структуру сложно создать, так как придётся долго копаться в коде или настраивать под неё CMS. Другой недостаток заключается в том, что при организации решётчатой структуры можно легко запутаться не только пользователю во время поиска информации, но и самому веб-мастеру при размещении контента.

На сайте с такой структурой у пользователя есть возможность перемещаться по веткам сайта не только вертикально (вверх-вниз), но и горизонтально (то есть между ветками на разных уровнях).

Основным недостатком такой структуры является большое количество гиперссылок, поэтому ее применение ограничено для больших сайтов. Применяется такая структура в основном для каталогов.

## Древовидная структура

Самой оптимальной и часто используемой структурой многими веб-мастерами в настоящий момент является древовидная структура. На обычном html её практически невозможно организовать (да и не нужно), поэтому для её создания пишется движок или используется CMS.

Главным ещё достоинством стоит отметить то, что она прекрасно может подойти для любого вида сайта, будь то домашняя веб-страница, сайт-визитка, корпоративный сайт, портал или каталог.

Смысл древовидной структуры заключается в том, что у человека на главной странице сайта есть выбор куда перейти: разделы, подразделы и конкретные документы. При переходе на любой уровень, пользователь может перейти в конец раздела, его начало, другой раздел или его подраздел, а также на главную страницу.

Всё это обеспечивает большую гибкость данной структуре. Однако при разрастании контента, практически невозможно будет соблюдать правильное соотношение между шириной и глубиной. В результате этого, посетителям, чтобы найти интересующую информацию, придётся сделать много лишних переходов по веб-страницам.

Более того, при разрастании контента поисковику будет всё сложнее индексировать содержание сайта. Тем не менее, древовидная структура - лучшее, что может подойти для любого сайта.

Древовидная структура — самый универсальный способ размещения web-страниц. Она подходит для создания практически любых типов сайтов. Ее принцип заключается в том, что пользователь при заходе на заглавную страницу

оказывается перед выбором, куда идти дальше. После перехода в нужный раздел, он подбирает необходимый подраздел и т.п. У древовидной структуры очень много достоинств, но так же есть и недостаток.

Речь идет о том, что в древовидной структуре очень сложно соблюдать баланс между глубиной и шириной. Если «дерево» сайта будет расти только вглубь, то пользователям, чтобы дойти до какой-то информации, придется загрузить и просмотреть слишком много страниц, что будет раздражать пользователей. А если создать очень широкую древовидную структуру, то посетители будут вынуждены каждый раз тратить много времени для выбора нужной им ветки. А это тоже плохо. Таким образом, при использовании древовидной структуры сайта необходимо постоянно следить за ее разрастанием и придерживаться золотой середины.

Такой тип структуры является универсальным для размещения веб-страниц. Принцип, такой структуры становится понятен уже из самого названия.

Каждая страница такой структуры, кроме главной страницы, является подразделом страниц более высокого уровня. Такая структура имеет начало — первую страницу, корень дерева, а конец просмотра связан с исчерпанием всего материала на сайте. Основным недостатком данного вида структуры является сложность соблюдения баланса между глубиной и шириной. Если дерево сайта будет расти только вглубь, то пользователю, чтобы дойти до интересующей его страницы, придется загрузить и просмотреть слишком большое количество страниц. Если же сайт будет только расширяться, то чтобы найти нужную информацию, пользователь будет тратить много времени на выбор нужного раздела. И первое, и второе вызывает негативные эмоции у пользователя, и чаще всего он покидает сайт разочарованным. Поэтому необходимо

постоянно следить за разрастанием ресурса и придерживаться золотой середины.

### **Смешанная структура**

Эта структура характеризуется, присутствием двух или более элементов выше перечисленных структур в одной, однако из-за сложности её реализации, она редко используется.

Эта структура уже на порядок сложнее всех рассмотренных ранее. В ней все страницы также размещаются в различных ветках. Но у пользователя есть возможность перемещаться по ним не только вертикально (вверх-вниз), но и горизонтально (то есть между ветками на разных уровнях). Используется решетка в основном только в каталогах. При этом перемещение между ветками на глубинных уровнях осуществляется с помощью отсылок на рубрики в других разделах.

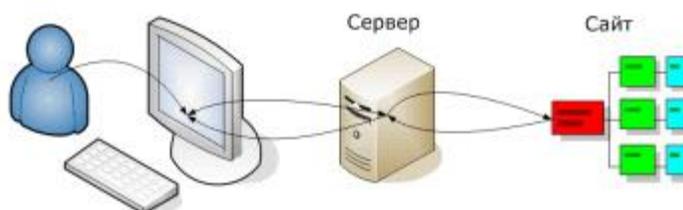
Использование решетчатой структуры в других проектах нецелесообразно. Во-первых, она относительно сложна в реализации. Во-вторых, обращаться с «решеткой» нужно с очень большой осторожностью, так как структура сайта может получиться очень запутанной, и посетители будут вынуждены долго блуждать в поисках нужной им информации.

### 1.3. Анализ технологий и методов построения веб сайтов

В сети Интернет существует огромное количество веб-сайтов. Все эти сайты можно разделить на два типа: **статические** и **динамические**.

#### Статические сайты

Статические сайты состоят из статических веб-страниц. Это означает, что независимо от действий пользователя страница всегда выглядит одинаково. Такие страницы хранятся на сервере как документы HTML. Верстаются статические страницы вручную. Если нужно изменить содержимое страницы, то приходится редактировать код HTML у каждой страницы.



#### Достоинства:

- Стоимостные затраты ниже;
- Минимальная нагрузка на веб-сервер.

#### Недостатки:

- Большие временные затраты на редактирование контента;
- Нецелесообразность использования в крупных проектах.

## Динамические сайты

Динамические сайты состоят из динамические веб-страницы, которые могут реагировать на действия пользователя и изменяться. Такие страницы формируются веб-сервером из нескольких файлов (шаблонов). Вся информация, как правило, хранится в базе данных. Когда пользователь запрашивает страницу, соответствующая информация извлекается из базы, вставляется в шаблон, образуя веб-страницу, и пересылается веб-сервером в пользовательский браузер. Таким образом, при обновлении содержимого сайта, необходимо просто добавить текст для новой страницы, который затем вставляется в базу данных с помощью определенного механизма. В результате получается, что сайт как бы сам себя обновляет.



## **2. Проектирование системы**

### **2.1. Требования к программе**

**Общие требования.** Данный сайт должен обеспечивать выполнение следующих основных операций:

- вывод информации на экран;
- запись и считывание информации в реальном времени;
- обработка введенной информации;
- ввод данных через форму.

**Требования к надежности.** Для предотвращения некорректной работы программы необходимо реализовать:

- семантический и синтаксический контроль исходных и вводимых данных;
- вывод сообщений об ошибках и возможность их устранения.

**Требования к условиям эксплуатации.** При работе с программой должны соблюдаться условия эксплуатации IBM PC- совместимых машин.

**Требования к составу и параметрам технических средств.** Для функционирования программного обеспечения необходимо наличие следующих технических средств:

- IBM PC-совместимая машина;
- клавиатура (для ввода информации);
- монитор (для вывода информации);
- Доступ к сети Интернет.

**Требования к информационной и программной совместимости.** Данный сайт должен представлять собой самостоятельный исполняемый модуль.

Сайт должна быть реализована с использованием одного из известных веб - языков программирования (HTML/PHP MYSQL.) и работать под управлением любой операционной системы.

**Требования к программной документации.** Для разрабатываемого сайта должна быть разработана следующая техническая документация:

- текст программы;
- сайт и методика испытаний данного ПО.

## **2.2. Стадии и этапы разработки**

**1. Технический проект.** На данном этапе проектирования должны быть определены структуры входных и выходных данных, а также форма их представления. Необходимо разработать алгоритм решения задачи и структуру сайта. Одновременно требуется определить семантику и синтаксис языка запросов.

Прежде всего, необходимо предварительно установить конфигурацию технических средств. Подготовить пояснительную записку.

**2. Рабочий проект.** На данном этапе необходимо провести программирование и отладку программного обеспечения, корректировку программной документации и методики испытаний. В то же время требуется подготовить контрольно-отладочные мероприятия и провести предварительные испытания разработанного сайта, выполнить модификацию сайта по итогам испытаний.

Необходимо окончательно оформить пояснительную записку и графический материал.

**3. Внедрение.** На данном этапе требуется подготовить и передать сайт и программную документацию для сопровождения. сайт должна теперь представлять собой законченный программный продукт.

### **2.3. Постановка задачи**

Сайт авто лизинговой компании предназначен для предоставления информации об автомобилях и способах выдачи её в лизинг.

Целью настоящей ВКР является разработка сайта в качестве информационного портала для авто лизинговой компании.

Данный сайт должен обеспечивать выполнение следующих основных операций:

- ввод с клавиатуры и корректировка исходных данных (добавление объявлений, их публикация и изменение);
- запись и считывание информации в реальном времени;
- обработка введенной информации (включает в себя обработку объявлений, их поиск и вывод на экран результатов поиска);
- вывод данных на дисплей, т.е. удобное и практичное для пользователя отображение информации, дружелюбный интерфейс программы.

С целью реализации поставленной задачи в данной работе я воспользовалась возможностями разных систем и технологий. Здесь перечислен список основных таких технологий:

- язык программирования PHP
- язык разметки HTML
- СУБД MySQL
- технологии Fancybox и CSS

- библиотека JQuery
- инструменты Bootstrap
- CMS Drupal

Теперь немного о каждой из этих технологий.

**PHP.** PHP (англ. PHP:Hypertext Preprocessor — «PHP: препроцессор гипертекста»; первоначально Personal Home Page Tools — «Инструменты для создания персональных веб-страниц») — скриптовый язык программирования общего назначения, интенсивно применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков программирования, применяющихся для создания динамических веб-сайтов.

Синтаксис PHP подобен синтаксису языка Си. Некоторые элементы, такие как ассоциативные массивы и цикл `foreach`, заимствованы из Perl.

Для работы программы не требуется описывать какие-либо переменные, используемые модули и т. п. Любая программа может начинаться непосредственно с оператора PHP.

PHP исполняет код, находящийся внутри ограничителей, таких как `<?php ?>`. Всё, что находится вне ограничителей, выводится без изменений.

```
<html>
  <head>
    <title> Авто лизнг </title>
  </head>
  <body>
    <?php echo 'Uzleasing.uz'; ?>
  </body>
</html>
```

**HTML.** HTML (от англ. HyperTextMarkupLanguage — «язык гипертекстовой разметки») — стандартный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами и отображается в виде документа в удобной для человека форме.

Язык HTML является приложением SGML (стандартного обобщённого языка разметки) и соответствует международному стандарту ISO 8879.

Язык XHTML является более строгим вариантом HTML, он следует всем ограничениям XML и, фактически, XHTML можно воспринимать как приложение языка XML к области разметки гипертекста.

Во всемирной паутине HTML-страницы, как правило, передаются браузерам от сервера по протоколам HTTP или HTTPS, в виде простого текста или с использованием сжатия.

Язык и его интерпретатор разрабатываются группой энтузиастов в рамках проекта с открытым кодом. Проект распространяется под собственной лицензией, несовместимой с GNU GPL.

Текстовые документы, содержащие разметку на языке HTML (такие документы традиционно имеют расширение .html или .htm), обрабатываются специальными приложениями, которые отображают документ в его форматированном виде. Такие приложения, называемые «браузерами» или «интернет-обозревателями», обычно предоставляют пользователю удобный интерфейс для запроса веб-страниц, их просмотра (и вывода на иные внешние устройства) и, при необходимости, отправки введённых пользователем данных на сервер. Наиболее популярными на сегодняшний день браузерами являются GoogleChrome, MozillaFirefox, Opera, InternetExplorer и Safari.

**Кроссбраузерность.** Одним из важных условий при разработке системы является её кроссбраузерность, поскольку система будет работать именно как веб-приложение.

Кроссбраузерность - это условие, которое говорит о том, что сайт одинаково и корректно отображается во всех современных браузерах, а так же их предыдущих версиях.

Проблема в том, что каждый разработчик браузера использует свои средства для обеспечения отображения того или иного свойства, что порождает необходимость разработчику сайта указывать одни и те же свойства по-разному для каждого типа браузеров. К счастью браузеры развиваются, и с каждым годом кроссбраузерность сайта становится все более простой задачей, но до совершенства еще далеко.

**Структура HTML-документа.** HTML — теговый язык разметки документов. Любой документ на языке HTML представляет собой набор элементов, причём начало и конец каждого элемента обозначается специальными пометками — тегами. Элементы могут быть пустыми, то есть не содержащими никакого текста и других данных (например, тег перевода строки <br>). В этом случае обычно не указывается закрывающий тег. Кроме того, элементы могут иметь атрибуты, определяющие какие-либо их свойства (например, размер шрифта для элемента font).

**CSS.** CSS (англ. CascadingStyleSheets — каскадные таблицы стилей) — формальный язык описания внешнего вида документа, написанного с использованием языка разметки.

Преимущественно используется как средство описания, оформления внешнего вида веб-страниц, написанных с помощью языков разметки HTML и

XHTML, но может также применяться к любым XML-документам, например, к SVG или XUL.

Применение CSS к документам HTML основано на принципах наследования и каскадирования. Принцип наследования заключается в том, что свойства CSS, объявленные для элементов-предков, наследуются элементами потомками. Но, естественно, не все свойства CSS наследуются — например, если для тега параграфа *p* средствами CSS задана рамка, то она не будет наследоваться ни одним тегом, содержащимся в данном теге *p*, а вот если для параграфа *p* средствами CSS задан цвет шрифта (например, `color:green;`), то это свойство будет унаследовано каждым элементом-тегом, находящимся в параграфе.

Принцип каскадирования применяется в случае, когда какому-то элементу HTML одновременно поставлено в соответствие более одного правила CSS, то есть, когда происходит конфликт значений этих правил. Чтобы разрешить такие конфликты вводятся правила приоритета.

С появлением CSS стало возможным принципиальное разделение содержания и представления документа. За счёт этого нововведения стало возможным лёгкое применение единого стиля оформления для массы схожих документов, а также быстрое изменение этого оформления.

**MySQL.** MySQL— свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

MySQL является решением для малых и средних приложений. Входит в состав серверов WAMP, AppServ, LAMP и в портативные сборки серверов Денвер, XAMP. Обычно MySQL используется в качестве сервера, к которому обращаются локальные или удалённые клиенты, однако в дистрибутив входит библиотека внутреннего сервера, позволяющая включать MySQL в автономные программы.

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. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц.

MySQL портирована на большое количество платформ: AIX, BSDi, FreeBSD, HP-UX, Linux, Mac OS X, NetBSD, OpenBSD, OS/2 Warp, SGI IRIX, Solaris, SunOS, SCO OpenServer, UnixWare, Tru64, Windows XP, WindowsServer 2003, WinCE, WindowsVista и Windows 7 и 8.

Важно отметить, что на официальном сайте СУБД для свободной загрузки предоставляются не только исходные коды, но и откомпилированные и оптимизированные под конкретные операционные системы готовые исполняемые модули СУБД MySQL.

MySQL имеет API для языков Delphi, C, C++, Эйфель, Java, Лисп, Perl, PHP, Python, Ruby, Smalltalk, Компонентный Паскаль и Tcl, библиотеки для языков платформы .NET, а также обеспечивает поддержку для ODBC посредством ODBC-драйвера MyODBC.

**Синтаксис.** Создание таблицы в базе данных производится командой  
CREATE TABLE.

```
CREATE [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name  
[(create_definition,...)]  
[table_options] [select_statement]
```

*tbl\_name* - Задаёт имя таблицы, которая будет создана в текущей базе данных. Если никакая база данных на момент вызова команды CREATE TABLE не была принята текущей, то возникнет ошибка выполнения команды. Начиная с MySQL 3.22 введена возможность явно указать базу данных, в которой будет создана новая таблица, при помощи синтаксиса *db\_name.tbl\_name*.

*TEMPORARY* - Этот параметр используется для создания временной таблицы с именем *tbl\_name* в течении только текущего сценария. По окончании выполнения сценария созданная таблица удаляется. Данная возможность появилась в MySQL 3.23 ВMySQL 4.0.2 для создания временных таблиц требуются привелегии CREATE TEMPORARY TABLES.

*IF NOT EXISTS* - Если указан этот параметр и производится попытка создать таблицу с дублирующим именем (т.е. таблица с таким именем в текущей БД уже есть), то таблица создана не будет и сообщение об ошибке не появиться. В противном случае таблица также создана не будет, но команда вызовет ошибку. Следует отметить, что при создании сравниваются только имена таблиц. Внутренние структуры не сравниваются.

*create\_definition* - Определяет внутреннюю структуру создаваемой таблицы (названия и типы полей, ключи, индексы и т.д.)

**jQuery.** jQuery — библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать

доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими. Также библиотека jQuery предоставляет удобный API для работы с AJAX.

Возможности:

- Движок кроссбраузерных CSS-селекторов Sizzle, выделенный в отдельный проект;
- Переход по дереву DOM, включая поддержку XPath как плагина;
- События;
- Визуальные эффекты;
- AJAX-дополнения;
- JavaScript-плагины.

Использование:

jQuery, как правило, включается в веб-страницу как один внешний JavaScript-файл:

```
<head>  
<script type="text/javascript" src="путьк jQuery.js"></script>  
</head>
```

Вся работа с jQuery ведётся с помощью функции \$. Если на сайте применяются другие JavaScript библиотеки, где \$ может использоваться для своих нужд, то можно использовать её синоним — jQuery. Второй способ считается более правильным, а чтобы код не получался слишком громоздким, можно писать его следующим образом:

```
jQuery(function($) {  
    // здесь код скрипта, где в $ будет находиться объект,  
    //предоставляющий доступ к функциям jQuery  
})  
)
```

FancyBox - это jQuery плагин, который служит для создания фотогалерей и вывода другого контента в красивых модальных окнах. Если планируется на сайте размещение некоторого контента в модальных окнах, этот скрипт окажет в таком случае огромную помощь.

Fancybox позволяет выводить в окне не только изображения, но и самый разнообразный контент: видео, iframe, HTML и др. Картинки можно прокручивать как кнопками вперед/назад, так и колесом мыши. Программа имеет несколько настроек режима вывода картинок, скорости перелистывания, вида надписей под картинками и др.

Фотогалерея FancyBox пользуется популярностью у разработчиков сайтов из-за простоты установки, приятного дизайна вывода галереи изображений, а также возможности работы с разнообразным контентом и поддержки большинством браузеров. Из недостатков можно отметить отсутствие вывода номера изображения типа "Изображение X из N", но возможно, в новых версиях эта функция будет добавлена.

**TwitterBootstrap.** TwitterBootstrap — свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML и CSS шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейсов, включая JavaScript расширения.

Bootstrap использует самые современные наработки в области CSS и HTML, поэтому необходимо быть внимательным при поддержке старых браузеров.

Основные инструменты Bootstrap:

Сетки — заранее заданные размеры колонок, которые можно сразу же использовать, например ширина колонки 140px относится к классу `.span2`, который можно использовать в CSS описании документа.

Шаблоны — Фиксированный или резиновый шаблон документа.

Типографика — Описания шрифтов, определение некоторых классов для шрифтов, таких как код, цитаты и т. п.

Медиа — Представляет некоторое управление изображениями и Видео.

Таблицы — Средства оформления таблиц, вплоть до добавления функциональности сортировки.

Формы — Классы для оформления не только форм, но и некоторых событий происходящих с ними.

Навигация — Классы оформления для Табов, Вкладок, Страничности, Меню и Тулбара.

Алерты — Оформление диалоговых окон, Подсказок и Всплывающих окон.

**Drupal** — система управления содержимым, используемая также как каркас для веб-приложений (CMF), написанная на языке PHP и использующая в качестве хранилища данных реляционную базу данных (поддерживаются MySQL, PostgreSQL и другие). Drupal является свободным программным обеспечением, защищённым лицензией GPL, и развивается усилиями энтузиастов со всего мира.

Начал разработку в 2000 году бельгиец Дрис Бёйтарт (Dries Buytaert), который и поныне является руководителем проекта. Название drupal является искажённым произношением голландского слова `druppel` (капля) и появилось в результате

забавной ошибки. В 2000 году несколько студентов Антверпенского университета использовали общий ADSL-модем для доступа в интернет, и для общения пользователей этой маленькой локальной сети Дрис создал внутренний сайт. Программа, на которой работал сайт, не имела имени, пока Дрис не собрался регистрировать доменное имя для нового сайта, и в результате опечатки вместо `dorp.org` (от голландского `dorp` — деревня, вполне уместное имя для небольшого сообщества) был зарегистрирован `drop.org` (англ. `drop` — капля). Аудитория нового сайта со временем сменилась на интересующихся развитием новых веб-технологий, и новый сайт постепенно перешёл из любительского эксперимента в проект по обсуждениям идей веб-программирования. В 2001 году Дрис опубликовал исходный код `drop.org` под именем «`Drupal`». С тех пор разработка переместилась на зарегистрированный позднее сайт `drupal.org`, который приобрёл статус официального сайта проекта. Со времени создания в процесс разработки были вовлечены сотни людей. Официальные разработчики ядра сейчас — это десятки человек, но много людей присылают свои патчи, включаемые в код системы. Также функциональность движка расширяется дополнительными модулями, которые создают сторонние разработчики. Например, в релизе 4.6 приняли участие 50 человек, приславших 338 поправок к коду, а в релизе 4.7 приняли участие уже 338 разработчиков, приславших примерно 1500 поправок. За пять лет существования проекта (к 2006 году) вышло 13 релизов. На данный момент (январь 2011) было разработано более 7000 дополнительных модулей, расширяющих базовые функции CMS. По статистике обращения к обновлениям движка на март 2009 года в интернете в пределах 120 тыс. активных сайтов на `Drupal 6` (без учёта сайтов находящихся за файрволлами, сайтов с отключённым механизмом обновлений). Это также не учитывает сайтов работающих на ранних версиях `Drupal`.

В 2007 году Дрис Бейтарт создал фирму Acquia, направленную на коммерческую поддержку проектов и сервисов, основанных на Drupal, вдохновленный успехами Canonical и RedHat, построивших бизнес на поддержке открытых проектов. В 2008 Бейтарт получил докторскую степень в области информатики и приборостроения («Computer Science and Engineering»). Со временем вокруг международного сайта появились национальные сообщества, функционируют более 20 национальных сайтов поддержки на разных языках.

архитектура Drupal позволяет применять его для построения различных типов сайтов — от блогов и новостных сайтов до информационных архивов или социальных сетей. Имеющуюся по умолчанию функциональность можно увеличивать подключением дополнительных расширений — «модулей» в терминологии Drupal. Наиболее важные функции, предоставляемые Drupal «из коробки»:

единая категоризация всех видов содержимого (таксономия) — от форумных сообщений до блогов и новостных статей

широкий набор свойств при построении рубрикаторов: плоские списки, иерархии, иерархии с общими предками, синонимы, родственные категории  
вложенность категорий любой глубины

поиск по содержимому сайта, в том числе поиск по таксономии и пользователям

разграничение доступа пользователей к материалам (ролевая модель)

динамическое построение меню

поддержка XML-форматов:

вывод документов в RDF/RSS

агрегация материалов с других сайтов

BlogAPI для публикации материалов с помощью внешних приложений

авторизация через OpenID

символьные осмысленные URL (иначе «человеко-понятные» — ЧПУ)

переводы интерфейса сайта на разные языки, а также поддержка ведения разноязычного контента

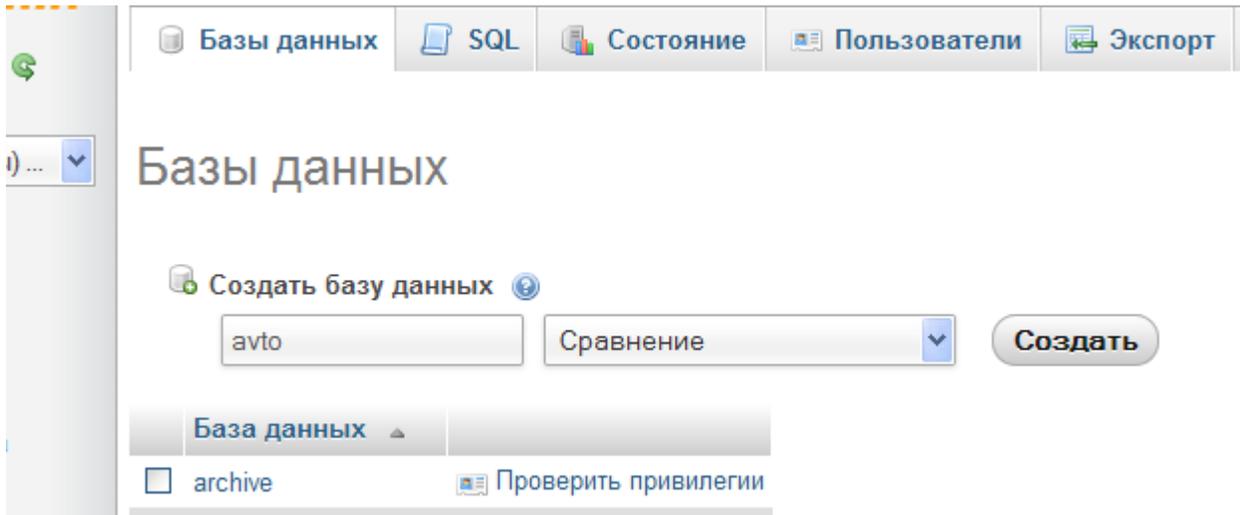
возможность создания сайтов с пересекающимся содержимым (например общей базой пользователей или общими настройками)

раздельные конфигурации сайта для различных виртуальных хостов (мультисайтинг), в том числе собственные наборы модулей и тем оформления для каждого подсайта.

уведомления о выходящих обновлениях модулей

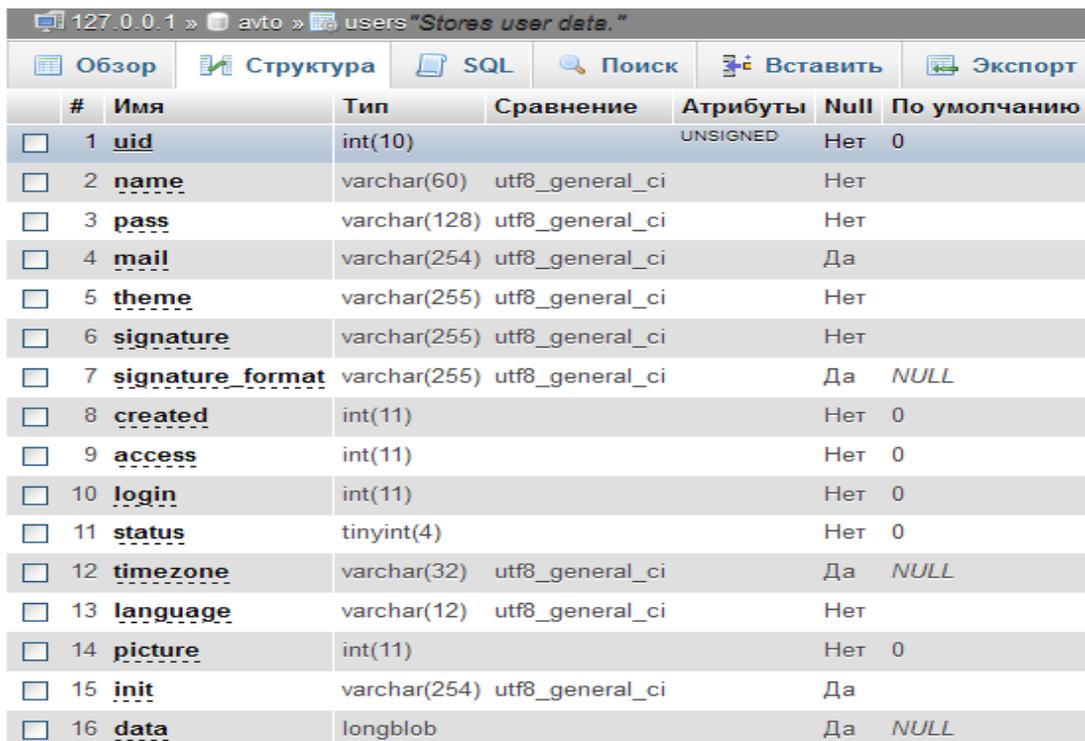
## 2.4. Структурная схема базы данных

Для работы Web-приложения необходима база данных, в которой хранятся сведения об объявлениях и пользователях. В соответствии с поставленной задачей была создана БД «avto».



### Таблица USERS:

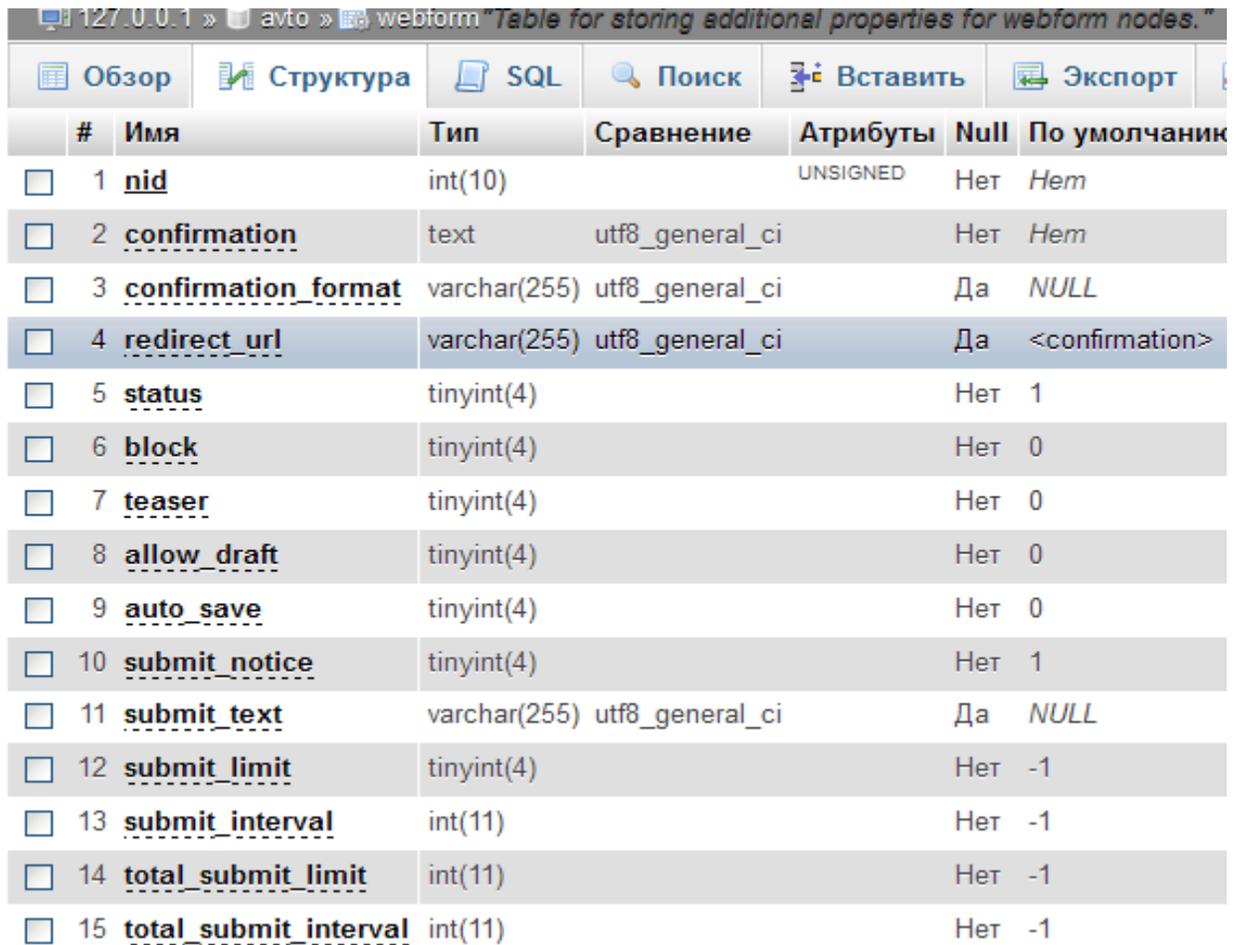
Данная таблица содержит сведения о пользователях.



#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
1	<u>uid</u>	int(10)		UNSIGNED	Нет	0
2	<u>name</u>	varchar(60)	utf8_general_ci		Нет	
3	<u>pass</u>	varchar(128)	utf8_general_ci		Нет	
4	<u>mail</u>	varchar(254)	utf8_general_ci		Да	
5	<u>theme</u>	varchar(255)	utf8_general_ci		Нет	
6	<u>signature</u>	varchar(255)	utf8_general_ci		Нет	
7	<u>signature_format</u>	varchar(255)	utf8_general_ci		Да	NULL
8	<u>created</u>	int(11)			Нет	0
9	<u>access</u>	int(11)			Нет	0
10	<u>login</u>	int(11)			Нет	0
11	<u>status</u>	tinyint(4)			Нет	0
12	<u>timezone</u>	varchar(32)	utf8_general_ci		Да	NULL
13	<u>language</u>	varchar(12)	utf8_general_ci		Нет	
14	<u>picture</u>	int(11)			Нет	0
15	<u>init</u>	varchar(254)	utf8_general_ci		Да	
16	<u>data</u>	longblob			Да	NULL

## Таблица WEBFORM:

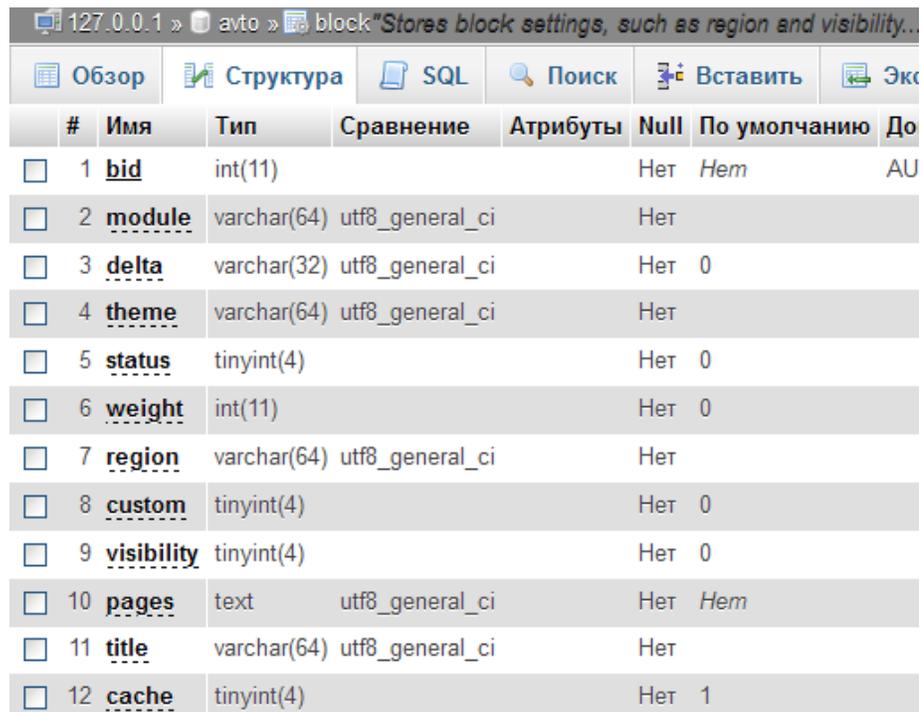
Данная таблица содержит сведения об формах.



#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию
<input type="checkbox"/>	1 <u>nid</u>	int(10)		UNSIGNED	Нет	Нет
<input type="checkbox"/>	2 <u>confirmation</u>	text	utf8_general_ci		Нет	Нет
<input type="checkbox"/>	3 <u>confirmation_format</u>	varchar(255)	utf8_general_ci		Да	NULL
<input type="checkbox"/>	4 <u>redirect_url</u>	varchar(255)	utf8_general_ci		Да	<confirmation>
<input type="checkbox"/>	5 <u>status</u>	tinyint(4)			Нет	1
<input type="checkbox"/>	6 <u>block</u>	tinyint(4)			Нет	0
<input type="checkbox"/>	7 <u>teaser</u>	tinyint(4)			Нет	0
<input type="checkbox"/>	8 <u>allow_draft</u>	tinyint(4)			Нет	0
<input type="checkbox"/>	9 <u>auto_save</u>	tinyint(4)			Нет	0
<input type="checkbox"/>	10 <u>submit_notice</u>	tinyint(4)			Нет	1
<input type="checkbox"/>	11 <u>submit_text</u>	varchar(255)	utf8_general_ci		Да	NULL
<input type="checkbox"/>	12 <u>submit_limit</u>	tinyint(4)			Нет	-1
<input type="checkbox"/>	13 <u>submit_interval</u>	int(11)			Нет	-1
<input type="checkbox"/>	14 <u>total_submit_limit</u>	int(11)			Нет	-1
<input type="checkbox"/>	15 <u>total_submit_interval</u>	int(11)			Нет	-1

## Таблица BLOCKS:

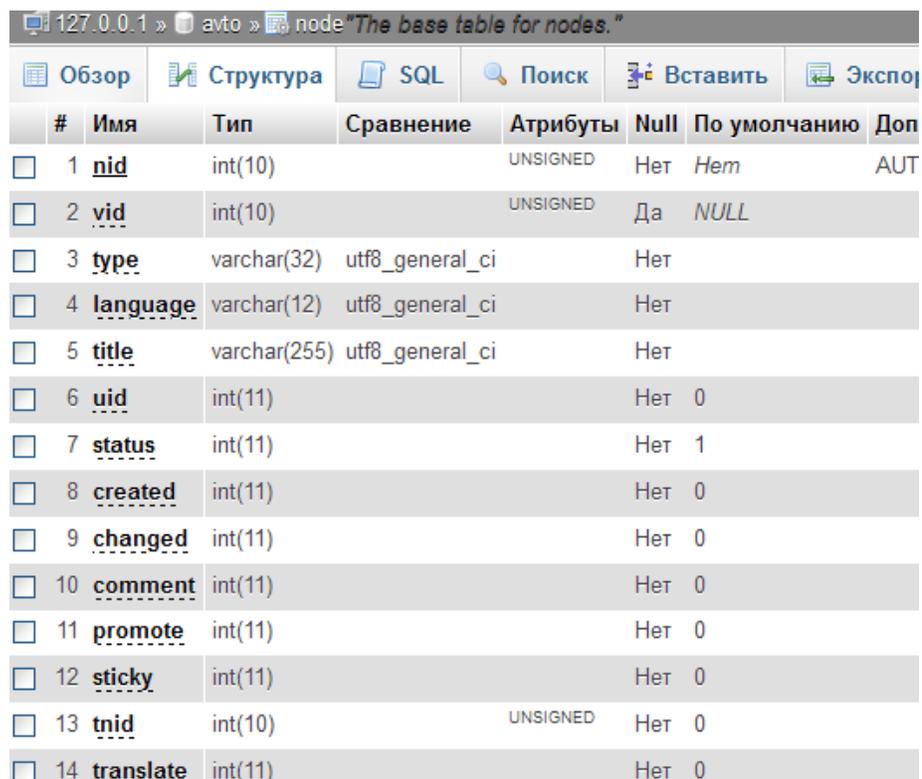
Данная таблица содержит сведения о блоках.



#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Доп
<input type="checkbox"/>	1 <u>bid</u>	int(11)			Нет	Нем	AU
<input type="checkbox"/>	2 <u>module</u>	varchar(64)	utf8_general_ci		Нет		
<input type="checkbox"/>	3 <u>delta</u>	varchar(32)	utf8_general_ci		Нет	0	
<input type="checkbox"/>	4 <u>theme</u>	varchar(64)	utf8_general_ci		Нет		
<input type="checkbox"/>	5 <u>status</u>	tinyint(4)			Нет	0	
<input type="checkbox"/>	6 <u>weight</u>	int(11)			Нет	0	
<input type="checkbox"/>	7 <u>region</u>	varchar(64)	utf8_general_ci		Нет		
<input type="checkbox"/>	8 <u>custom</u>	tinyint(4)			Нет	0	
<input type="checkbox"/>	9 <u>visibility</u>	tinyint(4)			Нет	0	
<input type="checkbox"/>	10 <u>pages</u>	text	utf8_general_ci		Нет	Нем	
<input type="checkbox"/>	11 <u>title</u>	varchar(64)	utf8_general_ci		Нет		
<input type="checkbox"/>	12 <u>cache</u>	tinyint(4)			Нет	1	

## Таблица NODES:

Данная таблица содержит сведения об отдельных страницах системы.



#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Доп
<input type="checkbox"/>	1 <u>nid</u>	int(10)		UNSIGNED	Нет	Нем	AUT
<input type="checkbox"/>	2 <u>vid</u>	int(10)		UNSIGNED	Да	NULL	
<input type="checkbox"/>	3 <u>type</u>	varchar(32)	utf8_general_ci		Нет		
<input type="checkbox"/>	4 <u>language</u>	varchar(12)	utf8_general_ci		Нет		
<input type="checkbox"/>	5 <u>title</u>	varchar(255)	utf8_general_ci		Нет		
<input type="checkbox"/>	6 <u>uid</u>	int(11)			Нет	0	
<input type="checkbox"/>	7 <u>status</u>	int(11)			Нет	1	
<input type="checkbox"/>	8 <u>created</u>	int(11)			Нет	0	
<input type="checkbox"/>	9 <u>changed</u>	int(11)			Нет	0	
<input type="checkbox"/>	10 <u>comment</u>	int(11)			Нет	0	
<input type="checkbox"/>	11 <u>promote</u>	int(11)			Нет	0	
<input type="checkbox"/>	12 <u>sticky</u>	int(11)			Нет	0	
<input type="checkbox"/>	13 <u>tnid</u>	int(10)		UNSIGNED	Нет	0	
<input type="checkbox"/>	14 <u>translate</u>	int(11)			Нет	0	

### 3. Описание веб-сайта

#### 3.1. Архитектура системы



Рисунок 3.1.1. Архитектура системы

Обобщенная архитектура сайта представлена в виде схемы на рисунке 3.1.1.

## 3.2. Навигация

При входе в систему пользователь попадает на главную страницу. В каком бы месте системы не находился пользователь, он всегда может вернуться на главную страницу (рисунок 3.2.1).

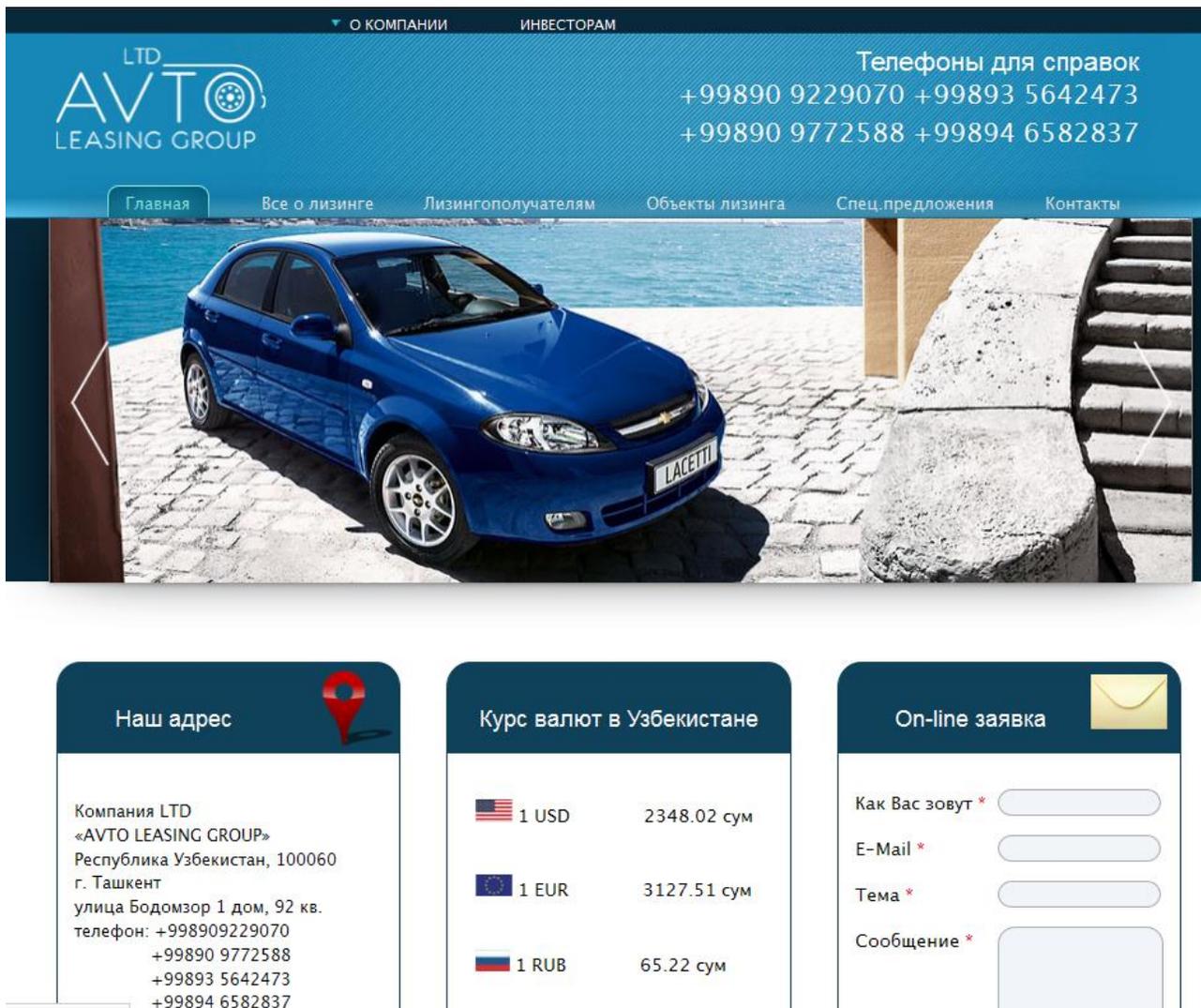
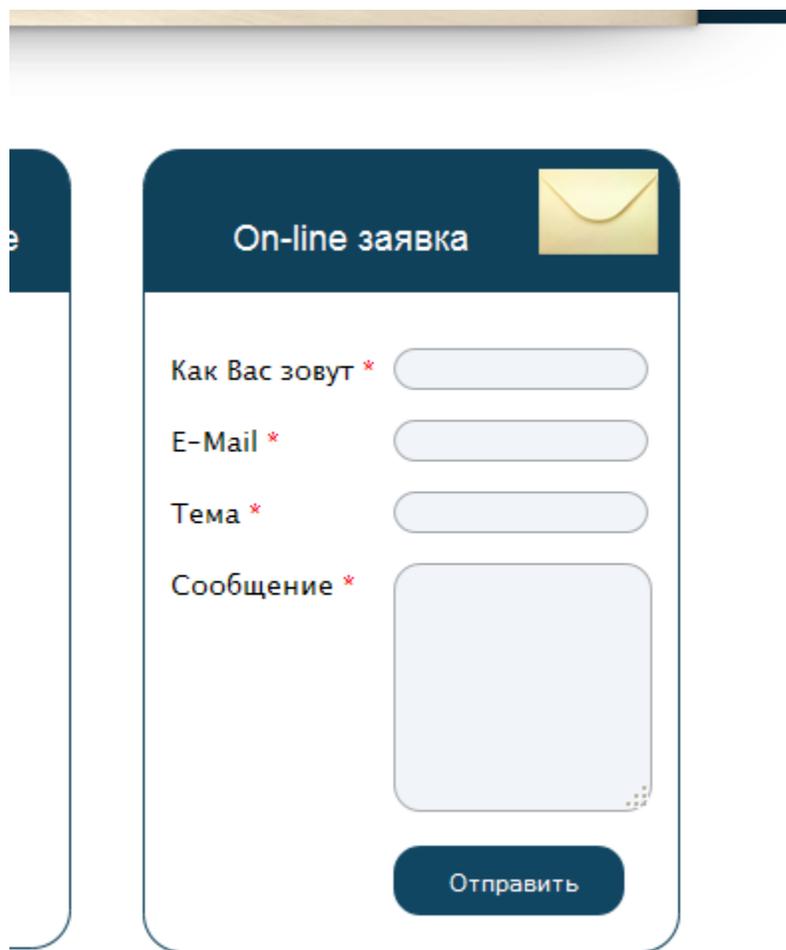


Рисунок 3.2.1. Главная страница сайта.

## 3.3. Онлайн заявка

Любой пользователь может находиться на сайте и искать нужный себе автомобиль. После нахождения оптимального для него варианта он может отправить онлайн-заявку (рисунок 3.3.1).



The image shows a mobile application interface for an online application form. The form is titled "On-line заявка" (Online application) and features a yellow envelope icon in the top right corner. The form contains four input fields, each with a red asterisk indicating a required field: "Как Вас зовут \*" (What is your name \*), "E-Mail \*" (E-Mail \*), "Тема \*" (Subject \*), and "Сообщение \*" (Message \*). The "Сообщение" field is a larger text area. At the bottom of the form is a dark blue button labeled "Отправить" (Send).

*Рисунок 3.3.1. Форма входа в систему*

### **3.3. Процесс добавления новых данных**

Для того чтобы добавить своё объявление, надо авторизоваться(рисунок 3.3.1).

Главная Печать Отправить по E-mail

Регистрация **Войти** Забыли пароль?

## Учётная запись пользователя

Имя пользователя \*

Укажите ваше имя на сайте Авто.

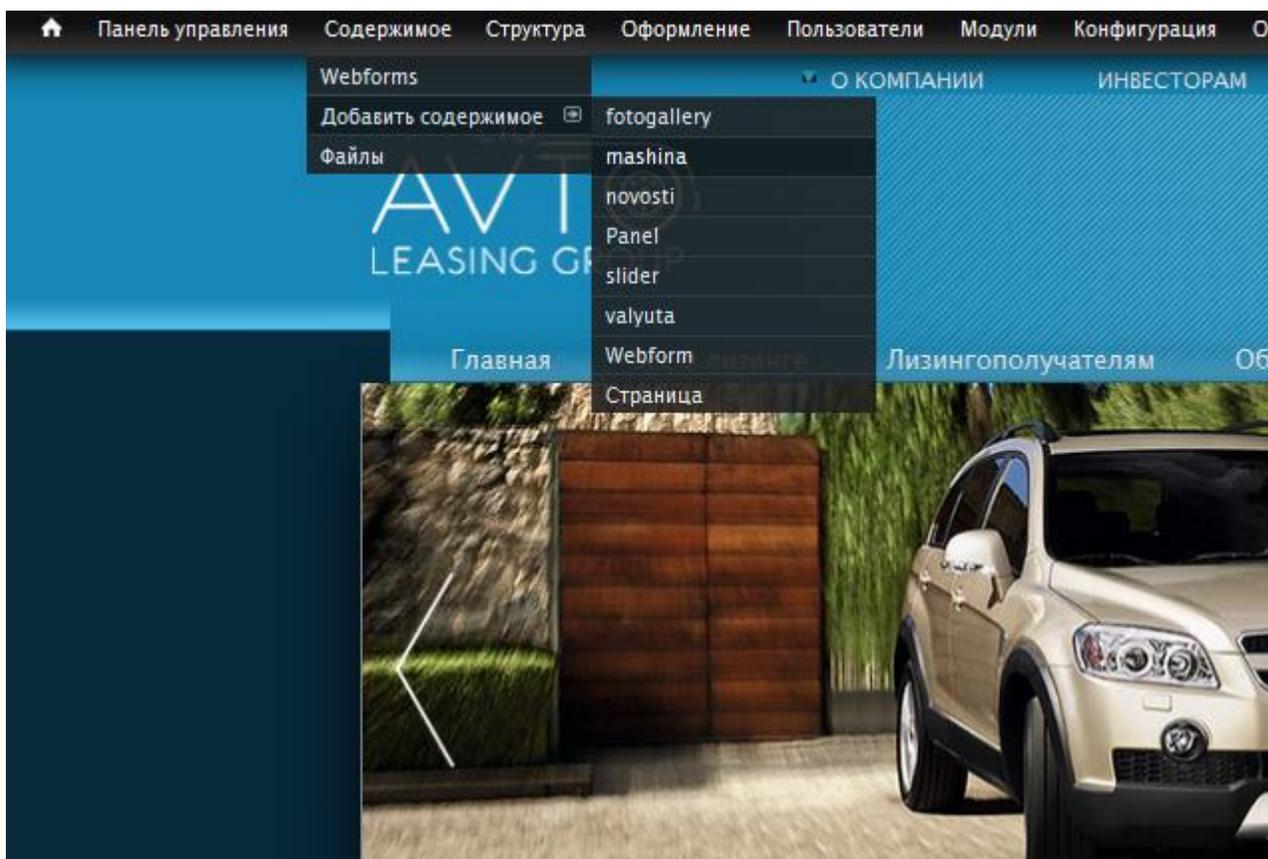
Пароль \*

Укажите пароль, соответствующий вашему имени пользователя.

Лизинг для физических лиц    Лизинг для юридических лиц    Лизинговые программы

*Рисунок 3.3.1. страница авторизации.*

После чего администратор сайта переходит в форму добавления нового продукта (рисунок 3.3.2):



*Рисунок 3.3.2. меню для перехода на страницу добавления продукта.*

## Создание материала mashina

Title \*

Body (Редактировать анонс)



body p

Disable rich-text

Формат текста Filtered HTML

- Адреса страниц и электронной почты автоматически преобразуются в ссылки.
  - Разрешённые HTML-теги: <a> <em> <strong> <cite> <blockquote> <code> <ul> <ol> <li> <dl> <dt> <dd>
- Строки и параграфы переносятся автоматически.

mash\_img

Обзор...  Закачать

Максимальный размер файла: 128 МБ.  
Разрешённые типы файлов: png gif jpg jpeg.

Тип

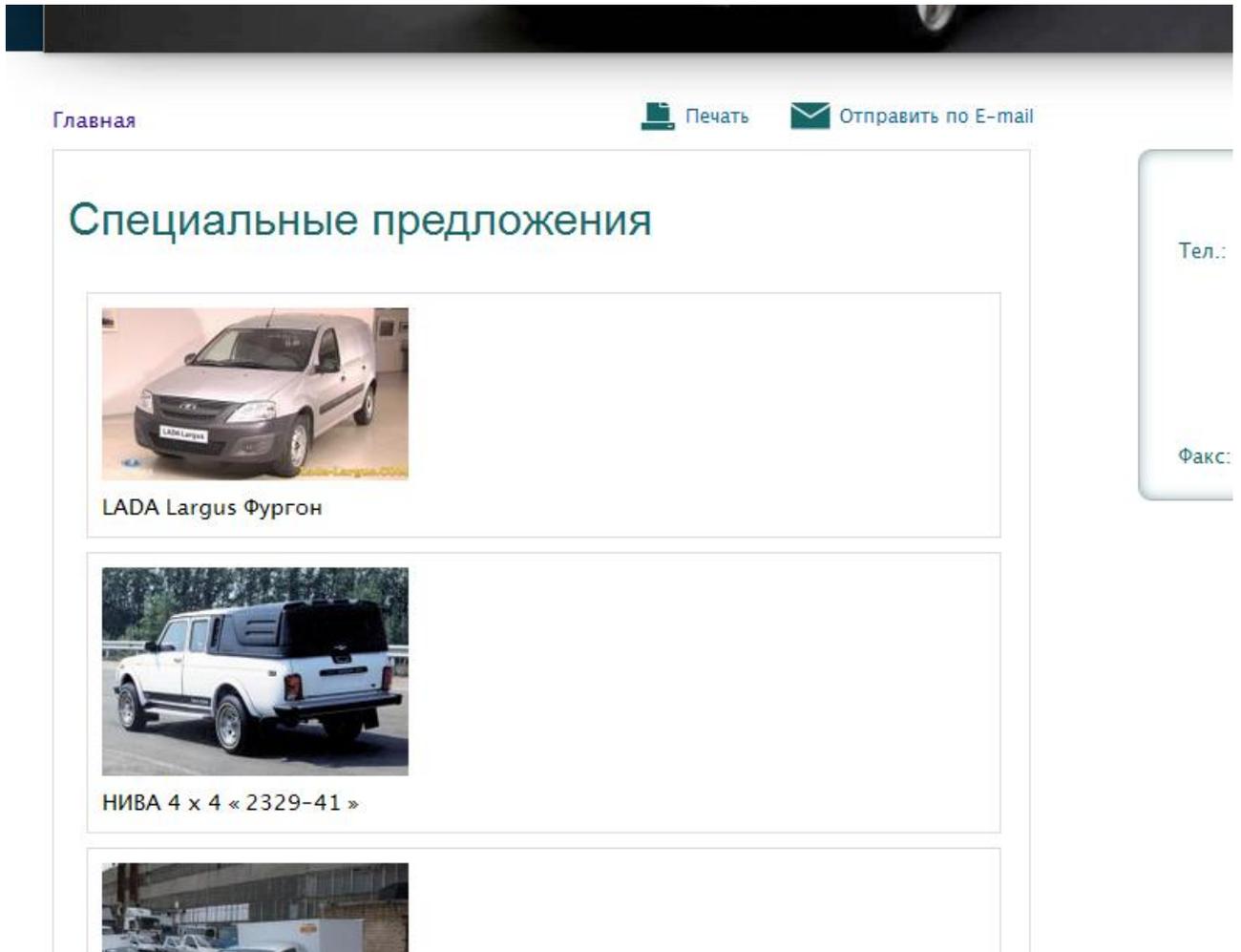
- Не указано -

*Рисунок 3.3.3. страница добавления продукта.*

После заполнения полей необходимо нажать на кнопку «Отправить». Данное действие означает, что данные попадут в БД системы и будут доступны для поиска по соответствующим критериям.

### 3.4. Просмотр предложений

Просмотр существующих предложений представлен в системе в виде списка с кратким описанием и возможностью перехода на просмотр расположения данного объекта на карте (рисунок 3.4.1).



*Рисунок 3.4.1. Список предложений*

После того, как пользователь нажмет на название или картинку автомобиля будет открыта страница с полной информацией о данном автомобиле (рисунок 3.4.2).

## Ново Грузовая бортовая машина ZZ1047D3414D145

Ново Грузовая бортовая машина ZZ1047D3414D145

Грузоподъемность - 1985 (5т.)

Размер : 5995\*2150\*2450

2014г. Цвет : Белый



Грузовые

*Рисунок 3.4.2. Просмотр полной информации об объекте.*

## Заключение

В ходе выполнения данной выпускной квалификационной работы были получены следующие результаты:

1. Рассмотрены возможности проектирования и построения сайтов;
2. Проведен анализ и выявлены положительные и отрицательные стороны используемого инструментария в проекте данной системы;
3. Разработана структура сайта;
4. Реализован проект сайта: получилась готовая к размещению в хостинг-ресурсах функционирующая система;
5. Рассмотрены основы безопасности жизнедеятельности, а именно порядок организации рабочего места, оснащенного компьютером и основы пожарной безопасности.

Веб-сайт ориентирован на широкий круг пользователей, который заинтересован в поиске автомобилей и других технических средств в лизинг. С её помощью пользователи могут искать выгодные предложения и просто интересующую их информацию о лизинге.

Данная сайт имеет возможность дальнейшего совершенствования через разработку и модернизацию модулей системы. Так же возможно развитие интерфейса с целью дальнейшего повышения его информативности, привлекательности и удобства для пользователей.

## Список использованной литературы

1. Постановление Президента Республики Узбекистан «О мерах по дальнейшему внедрению и развитию современных информационно-коммуникационных технологий» (Собрание законодательства Республики Узбекистан, 2012 г., № 13, ст. 139; 2013 г., № 44, ст. 578, № 45, ст. 584)
2. Экология и безопасность жизнедеятельности: Учебное пособие для студентов ВУЗов/ ред. Л. А. Муравий, 2002.
3. Белов С.В. Безопасность жизнедеятельности М.: Высшая школа. 2003.
4. Стивен Холзнер «РНР в примерах» изд. «Бином-Пресс», 2011 г
5. Теоретические основы безопасности жизнедеятельности. Айзман Р.И. и др. 2011, 208с.
6. Статьи с информационного интернет-ресурса <http://www.intuit.ru/>
7. Б. Д. Виснадул, С.А. Лупин, С.В Сидоров, П.Ю Чумаченко. Основы компьютерных сетей. (<http://library.tuit.uz>)
8. [http://ru.wikipedia.org/wiki/Twitter\\_Bootstrap](http://ru.wikipedia.org/wiki/Twitter_Bootstrap)
9. <http://ru.wikipedia.org/wiki/JQuery>
10. <http://ru.wikipedia.org/wiki/Html>
11. MySQL Reference Manuals с сайта <http://dev.mysql.com/doc/>
12. <http://ru.wikipedia.org/wiki/Css>
13. <http://ru.wikipedia.org/wiki/Mysql>
14. <http://ruseller.com/lessons.php?rub=32&id=238> Краткое руководство по библиотеке FancyBox
15. [http://seomark.ru/gallery\\_fancybox.html](http://seomark.ru/gallery_fancybox.html) Статья о галерее FancyBox
16. <https://ru.wikipedia.org/wiki/%D1%E0%E9%F2>
17. <https://ru.wikipedia.org/wiki/%D0%A1%D0%B0%D0%B9%D1%82-%D0%B2%D0%B8%D0%B7%D0%B8%D1%82%D0%BA%D0%B0>