Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра проектирования информационно-компьютерных систем

Дисциплина " Цель курсовой работы - разработка сайта визитки. Сайт-визитка содержит основную информацию об организации, частном лице, компании, товарах или услугах, прайс-листы, контактные данные и другую полезную информацию, что способствует привлечению новых покупателей и как следствие получение наибольшей выгоды. Получение наибольшей выгоды является основной задачей любого предприятия и предпринимателя, таким образом, внедрение сайта-визитки способствует развитию бизнеса. "

К защите допустить:
Руководитель курсовой работы

_____
13.07.2025

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

на тему

Разработка интернет-магазина по продаже автомобилей на фрэймворке Laravel

№ 21 ПЗ

Студент (подпись студента) А.В.Михалькевич

Курсовая работа
представлена на проверку
13.07.2025

_____
(подпись студента)

Минск 2025

Реферат

№ 21 ПЗ, гр.

А.В.Михалькевич, Разработка интернет-магазина по продаже автомобилей на фрэймворке Laravel, Минск: - 2025.

Пояснительная записка 277392 с., 4 рис., 0 табл.

Ключевые слова:

Предмет Цель курсовой работы - разработка сайта визитки. Сайт-визитка содержит основную информацию об организации, частном лице, компании, товарах или услугах, прайс-листы, контактные данные и другую полезную информацию, что способствует привлечению новых покупателей и как следствие получение наибольшей выгоды. Получение наибольшей выгоды является основной задачей любого предприятия и предпринимателя, таким образом, внедрение сайта-визитки способствует развитию бизнеса. ,

Содержание

Введение
1 Описание проекта
2 Обоснование выбора технологий
3 Инструментарий
4 Система контроля версий
5 Архитектурный шаблон проектирования HMVC
6 Разработка системы администрирования
Заключение
Список использованных источников
Приложения

Введение

Как правило, создание сайтов для коммерческих организаций является значимым. При этом они рассчитывают на его определенные функции. Во-первых, главным аспектом созданного сайта является повышение имиджа организации (имиджевая функция сайта). Во-вторых, сайт предоставляет всю необходимую информацию об услугах и продукции компании (информационная функция сайта). В-третьих, на сайте рекламируют продукцию и услуги компании (рекламная функция сайта). В-четвертых, сайт способствует привлечению новых клиентов и покупателей для увеличения продаж продукции и услуг (коммерческая функция сайта).

Цель данного курсового проекта: разработка веб-сайта интернет-магазина по продаже автомобилей.

Задачи: изучение современных инструментов разработки сайтов, реализация сайта-магазина, разработка баз данных для хранения информации.

1 Описание проекта

Курсовой проект представляет собой веб-сайт. Его основой является PHP-фреймворк Laravel. Для описания логической структуры веб-страницы используется язык разметки гипертекста HTML. Управление внешним видом документа и минимизация объема повторяющегося кода реализуется с помощью языка описания внешнего вида документа CSS. Также применяется язык сценариев JavaScript. Для хранения изменяющейся информации на сайте используются базы данных. Поддерживается возможность выбора товаров в корзину и их покупки. Реализованы возможности регистрации и авторизации, а также два режима прав доступа: администратор и пользователь.

2 Обоснование выбора технологий

Серверная часть

Для разработки серверной части веб-сайта был выбран язык PHP. PHP — один из популярных сценарных языков (наряду с JSP, Perl и языками, используемыми в ASP.NET).

Популярность в области построения веб-сайтов определяется наличием большого набора встроенных средств для разработки веб-приложений. Основные из них:

- автоматическое извлечение POST и GET-параметров, а также переменных окружения веб-сервера в предопределённые массивы;
- взаимодействие с большим количеством различных систем управления базами данных (MySQL, MySQLi, SQLite, PostgreSQL, Oracle (OCI8), Oracle, Microsoft SQL Server, Sybase, ODBC, mSQL, IBM DB2, Cloudscape и Apache Derby, Informix, Ovrimos SQL, Lotus Notes, DB++, DBM, dBase, DBX, FrontBase, FilePro, Ingres II, SESAM, Firebird / InterBase, Paradox File Access, MaxDB, Интерфейс PDO);
- автоматизированная отправка HTTP-заголовков;
- работа с HTTP-авторизацией;
- работа с cookies и сессиями;
- работа с локальными и удалёнными файлами, сокетами;
- обработка файлов, загружаемых на сервер;
- работа с XForms.

В настоящее время PHP используется сотнями тысяч разработчиков. Согласно рейтингу корпорации TIOBE, базирующемся на данных поисковых систем, в мае 2016 года PHP находился на 6 месте среди языков программирования. К крупнейшим сайтам, использующим PHP, относятся Facebook, Wikipedia и др.Входит в LAMP — распространённый набор программного обеспечения для создания и хостинга веб-сайтов (Linux, Apache, MySQL, PHP).

В частности, использовался фреймворк Laravel. Laravel — бесплатный веб-

фреймворк с открытым кодом, предназначенный для разработки с использованием архитектурной модели MVC (англ. Model View Controller — модель-представление-контроллер). Laravel выпущен под лицензией MIT. Исходный код проекта размещается на GitHub. В результате опроса sitepoint.com в декабре 2013 года о самых популярных PHP-фреймворках Laravel занял место самого многообещающего проекта на 2014 год.

В 2015 году в результате опроса sitepoint.com по использованию PHP-фреймворков среди программистов занял первое место в номинациях:

Фреймворк корпоративного уровня
Фреймворк для личных проектов

Ключевые особенности, лежащие в основе архитектуры Laravel:

1. Пакеты (англ. packages) — позволяют создавать и подключать модули в формате Composer к приложению на Laravel. Многие дополнительные возможности уже доступны в виде таких модулей.
2. Логика приложения — работает по шаблону проектирования HMVC.
3. Обратная маршрутизация связывает между собой генерируемые приложением ссылки и маршруты, позволяя изменять последние с автоматическим обновлением связанных ссылок. При создании ссылок с помощью именованных маршрутов Laravel автоматически генерирует конечные URL.
4. Автозагрузка классов — механизм автоматической загрузки классов PHP без необходимости подключать файлы их определений в include. Загрузка по требованию предотвращает загрузку ненужных компонентов; загружаются только те из них, которые действительно используются.
5. Составители представлений (англ. view composers) — блоки кода, которые выполняются при генерации представления (шаблона).
6. Инверсия управления (англ. Inversion of Control) — позволяет получать экземпляры объектов по принципу обратного управления. Также может использоваться для создания и получения объектов-одиночек (англ. singleton).
7. Миграции — система управления версиями для баз данных. Позволяет связывать изменения в коде приложения с изменениями, которые требуется внести в структуру БД, что упрощает развёртывание и обновление приложения.
8. Модульное тестирование (юнит-тесты) — играет очень большую роль в Laravel, который сам по себе содержит большое число тестов для предотвращения регрессий (ошибок вследствие обновления кода или исправления других ошибок).
9. Страничный вывод (англ. pagination) — упрощает генерацию страниц, заменяя различные способы решения этой задачи единым механизмом, встроенным в Laravel.

Для работы с базами данных использовался самый популярный (по последним опросам) в этой области язык – MySql.

Клиентская часть

Для клиентской части веб-сайта использовались языки разметки HTML и CSS, а также же скриптовый язык JavaScript – обязательные технологии для разработки frontend части любого веб-приложения. Также использовались:

Bootstrap (фреймворк) - свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

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. Сейчас разработка jQuery ведётся командой jQuery во главе с Джоном Резигом.

Внешние шрифты

3 Инструментарий

Open Server

Open Server — это портативная серверная платформа и программная среда, созданная специально для веб-разработчиков с учётом их рекомендаций и пожеланий. Программный комплекс имеет богатый набор серверного программного обеспечения, удобный, многофункциональный продуманный интерфейс, обладает мощными возможностями по администрированию и настройке компонентов. Платформа широко используется с целью разработки, отладки и тестирования веб-проектов, а также для предоставления веб-сервисов в локальных сетях. Хотя изначально программные продукты, входящие в состав комплекса, не разрабатывались специально для работы друг с другом, такая связка стала весьма популярной среди пользователей Windows, в первую очередь из-за того, что они получали бесплатный комплекс программ с надежностью на уровне Linux серверов. Удобство и простота управления безусловно не оставят вас равнодушными, за время своего существования Open Server зарекомендовал себя как первоклассный и надёжный инструмент необходимый каждому веб-мастеру.

В частности, очень удобен для работы с базами данных PhpMyAdmin со встроенным MySql. Использовались следующие настройки:

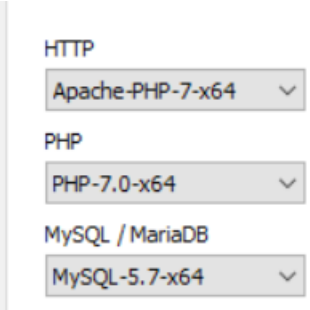


Рисунок 1 – Настройки Open Server

GitHub

GitHub — крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub, Inc (ранее Logical Awesome). Сервис абсолютно бесплатен для проектов с открытым исходным кодом и предоставляет им все возможности (включая SSL), а для частных проектов предлагаются различные платные тарифные планы. Слоган сервиса — «Social Coding» — на русский можно перевести как «Пишем код вместе». На футболках же печатают совсем другую фразу: «Fork you!» («Ответвись!»). С одной стороны, она созвучна с англоязычным ругательством и намекает на неформальную атмосферу. С другой, эти слова напоминают, что создавать новые форки с Git можно легко и безболезненно — традиционно, к созданию веток разработчики проектов с открытым исходным кодом относятся негативно — а также созвучна названию одной из возможностей Github — очереди форков.

Brackets

Brackets — свободный текстовый редактор для веб-разработчиков. Brackets ориентирован на работу с HTML, CSS и JavaScript. Эти же технологии лежат в основе самого редактора, что обеспечивает его кроссплатформенность т.е. совместимость с операционными системами Mac, Windows и Linux. Brackets создан и развивается Adobe Systems под лицензией MIT License и поддерживается на GitHub. На сегодняшний день сообществом создано множество расширений, добавляющих большинство необходимых инструментов для работы над кодом, таких как система контроля версий Git, просмотр HTML-кода в браузере в реальном времени (Live Preview), синхронизация с FTP (Git-FTP). Принять участие в разработке и поддержке расширений может любой желающий.

4 Система контроля версий

Git — распределённая система управления версиями. Проект был создан Линусом Торвальдсом для управления разработкой ядра Linux, первая версия выпущена 7 апреля 2005 года. На сегодняшний день его поддерживает Джунио Хамано. Среди проектов, использующих Git — ядро Linux, Swift, Android, Drupal, Cairo, GNU Core Utilities, Mesa, Wine, Chromium, CompizFusion, FlightGear, jQuery, PHP, NASM, MediaWiki, DokuWiki, Qt, ряд дистрибутивов Linux.

Устанавливается Git на Windows очень просто – при помощи инсталлятора. После установки есть возможность пользоваться SSH клиентом и стандартной графической версией. Далее создаётся локальный репозиторий. Для этого нужно зайти в папку с проектом и прописать в командной строке

«git init»

Для добавления всех файлов каталога под версионный контроль нужно набрать сначала добавить файлы командой

*«git add *»*

а после создать коммит при помощи

«git commit –m “commit comment here”»

После этих манипуляций можно перенести каталог в онлайн-репозиторий при помощи команды

«git push origin master имя_репозитория»

В нашем курсовом проекте использовался онлайн-репозиторий GitHub. Ссылка на него: https://github.com/pivchenkosv/socset1

5 Архитектурный шаблон проектирования HMVC

Концепция MVC (Model-View-Controller: модель-вид-контроллер) очень часто упоминается в мире веб программирования в последние годы. Каждый, кто хоть как-то связан с разработкой веб приложений, так или иначе сталкивался с данным акронимом. Сегодня мы разберёмся, что такое - концепция MVC, и почему она стала популярной. MVC — это не шаблон проекта, это конструкционный шаблон, который описывает способ построения структуры нашего приложения, сферы ответственности и взаимодействие каждой из частей в данной структуре. Впервые она была описана в 1979 году, конечно же, для другого окружения. Тогда не существовало концепции веб приложения. Tim Berners Lee (Тим Бернерс Ли) посеял семена World Wide Web (WWW) в начале девяностых и навсегда изменил мир. Шаблон, который мы используем сегодня, является адаптацией оригинального шаблона к веб разработке. Бешеная популярность данной структуры в веб приложениях сложилась благодаря её включению в две среды разработки, которые стали очень популярными: Struts и Ruby on Rails. Эти две среды разработки наметили пути развития для сотен рабочих сред, созданных позже. Идея, которая лежит в основе конструкционного шаблона MVC, очень проста: нужно чётко разделять ответственность за различное функционирование в наших приложениях:

Рисунок 2 – Разделение ответственности

Приложение разделяется на три основных компонента, каждый из которых отвечает за различные задачи. Давайте подробно разберём компоненты на примере:

Контроллер управляет запросами пользователя (получаемые в виде запросов HTTP GET или POST, когда пользователь нажимает на элементы интерфейса для выполнения различных действий). Его основная функция — вызывать и координировать действие необходимых ресурсов и объектов, нужных для выполнения действий, задаваемых пользователем. Обычно контроллер вызывает соответствующую модель для задачи и выбирает подходящий вид.

Модель - это данные и правила, которые используются для работы с данными, которые представляют концепцию управления приложением. В любом приложении вся структура моделируется как данные, которые обрабатываются определённым образом. Что такое пользователь для приложения — сообщение или книга? Только данные, которые должны быть обработаны в соответствии с правилами (дата не может указывать в будущее, e-mail должен быть в определённом формате, имя не может быть длиннее X символов, и так далее). Модель даёт контроллеру представление данных, которые запросил пользователь (сообщение, страницу книги, фотоальбом, и тому подобное). Модель данных будет одинаковой, вне зависимости от того, как мы хотим представлять их пользователю. Поэтому мы выбираем любой доступный вид для отображения данных. Модель содержит наиболее важную часть логики нашего приложения, логики, которая решает задачу, с которой мы имеем дело (форум, магазин, банк, и тому подобное). Контроллер содержит в основном организационную логику для самого приложения (очень похоже на ведение домашнего хозяйства).

Вид обеспечивает различные способы представления данных, которые получены из модели. Он может быть шаблоном, который заполняется данными. Может быть несколько различных видов, и контроллер выбирает, какой подходит наилучшим образом для текущей ситуации.

Веб приложение обычно состоит из набора контроллеров, моделей и видов. Контроллер может быть устроен как основной, который получает все запросы и вызывает другие контроллеры для выполнения действий в зависимости от ситуации.

HMVC - это эволюция концепции MVC, которая используется в многих веб приложениях. Она появилась как решение некоторых проблем, проявившихся при использовании MVC в веб приложениях. Решение было представлено на сайте JavaWorld в июле 2000. Предлагалось использовать стандартную триаду Модель-Контроллер-Вид использовать в качестве слоев в «**иерархии родитель-потомок**». Рисунок отображает принцип работы:

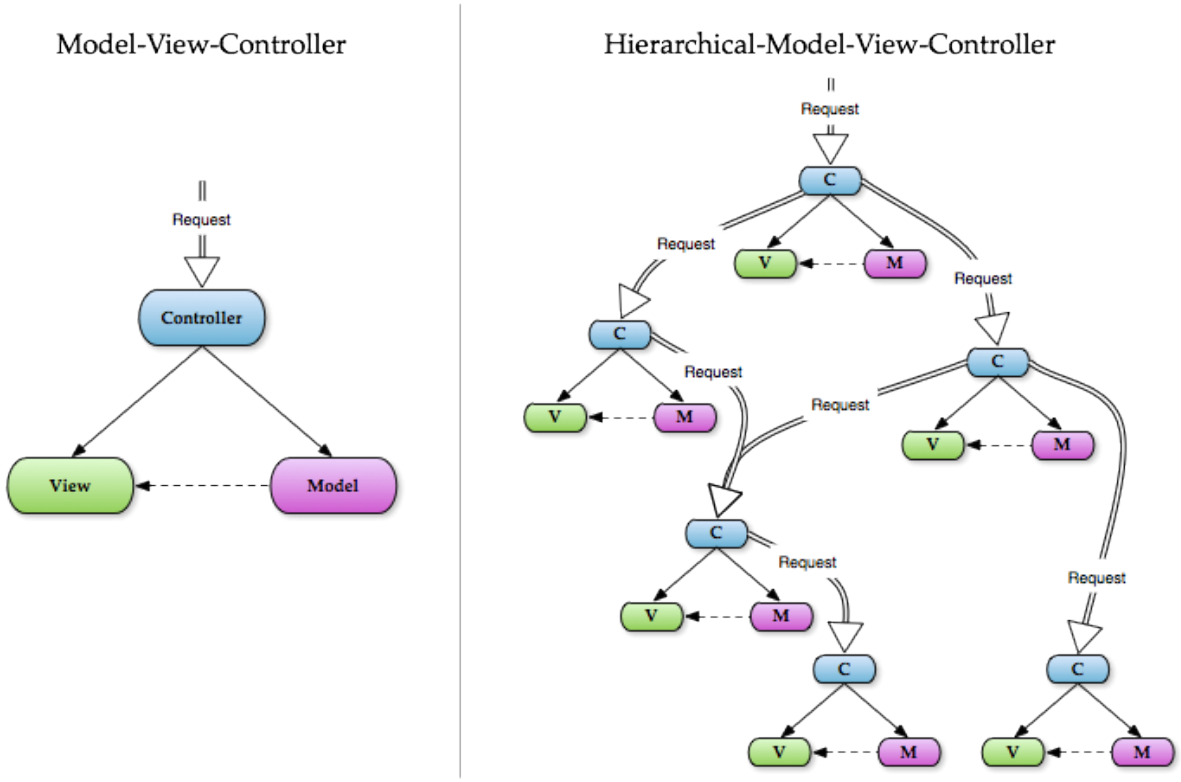


Рисунок 3 – Принцип работы HMVC

Каждая триада функционирует независимо от других. Триада может запросить доступ к другой триаде через их контроллеры. Обе точки позволяют приложению распределяться по нескольким местам, если нужно. В дополнение, использование слоев триад MVC позволяет добиться более глубокой и тщательной разработки приложений. Такой подход позволяет получить ряд преимуществ, которые будут описаны далее по тексту. Почему следует использовать HMVC? Ключевыми преимуществами, которые дает использование HMVC при разработке приложения, являются:

Модульность: Уменьшается зависимость между различными частями приложения.

Организация: Наличие папки для каждой значимой триады облегчает работу по загрузке приложения на сервер сайта.

Повторное использование: В следствии природы дизайна приложения, очень просто повторно использовать практически каждый кусок кода.

Расширяемость: Делает приложение доступным для расширения без жертвования легкостью поддержки.

Данные преимущества позволяют делать более совершенные приложения при снижении сложности разработки.

Маршрутизация (англ. Routing) — процесс определения маршрута следования данных в сетях связи. Маршруты могут задаваться административно (статические маршруты), либо вычисляться с помощью алгоритмов маршрутизации, базируясь на информации о топологии и состоянии сети, полученной с помощью протоколов маршрутизации (динамические маршруты).

Статическими маршрутами могут быть:

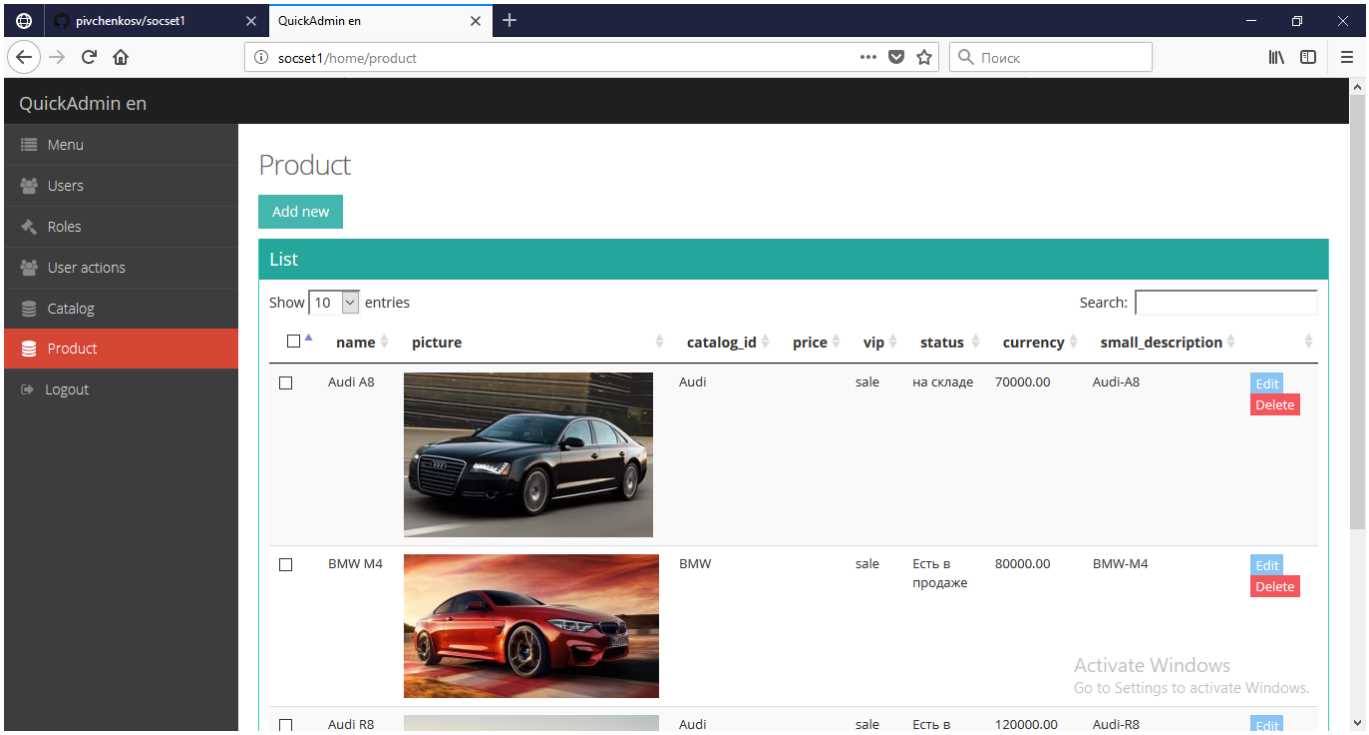
маршруты, не изменяющиеся во времени;

маршруты, изменяющиеся по расписанию;

Маршрутизация в компьютерных сетях выполняется специальными программно-аппаратными средствами — маршрутизаторами; в простых конфигурациях может выполняться и компьютерами общего назначения, соответственно настроенными.

6 Разработка системы администрирования

Готовая система администрирования выглядит так:



Сперва создаем авторизацию. Модуль авторизации встроен в Laravel. Для этого необходимо в консоле прописать следующую команду:

php artisan make:auth

Также необходимо обновить менеджер зависимостей, командой

composer self-update

Весь дальнейший процесс разработки системы администрации берем из документации Laravel Quickadmin, которая доступна по ссылке

https://laraveldaily.com/packages/quickadmin/

Первую команду из документации установки Laravel запускать не надо, т.к. проект Laravel уже создан.

Заключение

По ходу курсового проекта был создан веб-сайт для интернет-магазина по продаже автомобилей с возможностью добавления товаров в корзину и их удаления. Предусмотрены возможности регистрации и авторизации новых пользователей. Каталог товаров представляет собой разделы (фирмы-производители), в каждом из которых можно найти конкретный модели автомобилей. Задача была реализована при использовании PHP-фреймворка Laravel, языка разметки html, языка описания внешнего вида CSS и языка сценариев JavaScript. Также проект был перенесён на онлайн-репозиторий GitHub.

Список использованных источников

Приложения

1. [title] 5b3a72c4f2d00_title.docx
2. [picture] 5b3a751d6b3fa_pic_admin.jpg