

Министерство науки и высшего образования Российской Федерации
Муромский институт (филиал)
федерального государственного бюджетного образовательного учреждения высшего образования
«Владимирский государственный университет
Имени Александра Григорьевича и Николая Григорьевича Столетовых»
(МИВлГУ)

Факультет _____ ИТ _____

Кафедра _____ ПИИ _____

КУРСОВАЯ РАБОТА

по _____ WEB-технологиям _____

Тема _____ сайт фотографа _____

(оценка)

Руководитель

Рыбкин И.А.
(фамилия, инициалы)

(подпись) (дата)

Члены комиссии

Студент ПИИ-119
(группа)

(подпись) (Ф.И.О.)

Лямина И.А.
(фамилия, инициалы)

(подпись) (Ф.И.О.)

(подпись) (дата)

Задание

Задание

В данной курсовой работе нужно разработать сайт фотографа. Все необходимые данные для проекта будут находиться на локальном сервере OpenServer. В качестве базы данных для хранения необходимой информации выбрана MySql. Для создания сайта были также использованы следующие средства: редактор исходного кода Sublime Text 3, язык гипертекстовой разметки HTML и каскадные таблицы стилей CSS, языки программирования PHP и JavaScript.

In this course work, you need to develop a photogrof website. All necessary data for the project will be located on the local OpenServer server. MySql was chosen as the database for storing the necessary information. The following tools were also used to create the site: the Sublime Text 3 source code editor, the HTML hypertext markup language and CSS cascading style sheets, the PHP and JavaScript programming languages.

Содержание

Введение	6
1. Анализ технического задания	7
2. Описание языка PHP и JavaScript.....	9
2.1 Язык программирования PHP	9
2.2 Язык программирования JavaScript	11
3. Выбор и обоснование архитектуры системы (OpenServer).....	13
4. Выбор и обоснование алгоритмов	15
4.1 Алгоритм работы с cookie файлами	15
4.2 Алгоритм обработки форм.....	15
4.3 Алгоритм работы с БД.....	16
5. Описание программы.....	18
5.1 Описание сайта	18
5.2 Описание админ-панели	24
5.3 Структура базы данных.....	27
Заключение.....	29
Список используемой литературы.....	30
Приложение 1. Программный код веб-приложения	31

					МИВУ 09.03.04-8.000 ПЗ		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.		Лямина И.А.			Сайт фотографа	Лит.	Лист
Провер.		Рыбкин И.А.					Листов
Реценз.							5
Н. Контр.							44
Утверд.						МИ ВлГУ ПИН-119	

Введение

Интернет развивается очень быстро. Число публикаций, посвященных Всемирной паутине, быстро растет, что свидетельствует о ее широком использовании даже в отдаленных от технологии областях. Интернет превращается в полноценный источник разнообразной полезной информации для каждой категории пользователей.

Сейчас, чтобы привлечь внимание клиентов, покупателей или партнёров, необходимо заявить о себе в интернете, путём создания WEB-сайта. Для этих целей как раз и служит web-сайт, содержащий основную информацию об организации, частном лице, компании, товарах или услугах, прайс-листы, контактные данные. Сайты позволяют хранить, передавать, продавать различные типы информации, не отходя от экрана компьютера.

Целью курсовой работы является создание сайта фотографа.

Актуальность разработки сайта заключается в получении практических и теоретических навыков по созданию полноценного веб-приложения.

Задачи работы:

1. провести анализ предметной области;
2. спроектировать базу данных, где будет храниться основная информация;
3. продумать основные страницы сайта (что на них будет и как будут выглядеть);
4. разработать сайт.

					МИВУ 09.03.04-8.000 ПЗ	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

1. Анализ технического задания

В данной курсовой работе необходимо спроектировать и разработать сайт фотографа, где будет представлена основная информация о нем. На главной странице будет находиться описание. Будут созданы страницы блога, отзывов и возможность написания сообщения фотографу.

Проект будет храниться и разрабатываться на локальном сервере OpenServer. В проекте будут использоваться: язык PHP для взаимодействия с базой данных, JavaScript для различных графических эффектов и язык гипертекстовой разметки HTML с каскадными таблицами стилей CSS, которые служат для описания внешнего вида.

В соответствии с заданием на курсовую работу необходимо будет реализовать следующие пункты:

- 1) использование локального сервера (Apache, PHP, MySQL и т.д.);
- 2) создать базу данных текстов, информации об изображениях и разделах;
- 3) использовать загрузку и выгрузку информации из базы данных на страницу сайта, при помощи языка PHP и/или JavaScript;
- 4) создать кроссбраузерные, блочные шаблоны страниц (не менее трех шаблонов);
- 5) Динамическое изменение внешнего вида и/или информации на странице сайта;
- 6) Использование комментариев к функциям динамического ввода/вывода.

Для создания сайта необходимо использовать локальный сервер – это программа, создающая на ПК среду полноценного веб-хостинга. Необходимо это для того чтобы создать сайт у себя на компьютере, и в дальнейшем перенести его на хостинг - комплекс услуг, который позволяет разместить сайт в сети Интернет. Цели размещения сайта на хостинге в данной курсовой

					МИВУ 09.03.04-8.000 ПЗ	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

работе не имеется, поэтому все данные будут располагаться в папках локального сервера OpenServer.

MySQL представляет собой очень популярную систему управления базами данных с открытыми исходными текстами. В базе данных будут храниться различные типы данных, таких как int, varchar, text. Изображения не будут храниться в базе данных, только путь к ним, так как их извлечение сложная задача и требует немало памяти.

					МИВУ 09.03.04-8.000 ПЗ	Лист
						8
Изм.	Лист	№ докум.	Подпись	Дата		

2. Описание языка PHP и JavaScript

PHP – это распространенный язык программирования общего назначения с открытым исходным кодом. PHP специально сконструирован для веб-разработок и его код может внедряться непосредственно в HTML. Этот язык позволяет изменять веб-страницы на сервере непосредственно перед тем, как она будет отправлена браузеру.

JavaScript — это язык управления сценариями просмотра гипертекстовых страниц Web на стороне клиента. Если быть более точным, то JavaScript — это не только язык программирования на стороне клиента. Однако наибольшую популярность JavaScript обеспечило программирование на стороне клиента.

Основная идея JavaScript состоит в возможности изменения значений атрибутов HTML-контейнеров и свойств среды отображения в процессе просмотра HTML-страницы пользователем. При этом перезагрузки страницы не происходит.

PHP выполняется на стороне сервера, в то время как JavaScript выполняется в браузере на стороне пользователя. Язык JavaScript полностью объектно-ориентирован, тогда как в языке PHP объекты рассматриваются как необязательное средство. Такое различие систем обозначений имеет свое преимущество в том, что невозможно когда-либо спутать объект JavaScript с объектом PHP, или наоборот. Недостатком является то, что отсутствует возможность обратиться к одному и тому же объекту из обоих языков.

2.1 Язык программирования PHP

PHP — 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.

В области веб-программирования, в частности серверная часть, PHP — один из популярных сценарных языков (наряду с JSP, Perl и языками, используемыми в ASP.NET) благодаря своей простоте, скорости выполнения, богатой функциональности, кроссплатформенности и распространению исходных кодов на основе лицензии PHP.

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 ТЮВЕ, базирующемуся на данных поисковых систем, в июне 2013 года PHP находился на 5 месте среди языков программирования. К крупнейшим сайтам, использующим PHP, относятся Facebook, Wikipedia и др.

2.2 Язык программирования JavaScript

JavaScript - предназначен для написания сценариев для активных HTML-страниц. JavaScript не предназначен для создания автономных приложений. Программа на JavaScript встраивается непосредственно в исходный текст HTML-документа и интерпретируется браузером по мере загрузки этого документа. С помощью JavaScript можно динамически изменять текст загружаемого HTML-документа и реагировать на события, связанные с действиями посетителя или изменениями состояния документа или окна.

Важная особенность JavaScript - объектная ориентированность. Программисту доступны многочисленные объекты, такие, как документы, гиперссылки, формы, фреймы и т.д. Объекты характеризуются описательной информацией (свойствами) и возможными действиями (методами).

Достоинства и недостатки

Javascript популярен не случайно, а благодаря своим несомненным достоинствам.

— Незаменимость для веб-разработки. Поддержка скриптов всеми популярными браузерами; полная интеграция с вёрсткой страниц (HTML+CSS) и серверной частью (backend).

— Скорость работы и производительность. Javascript позволяет частично обрабатывать веб-страницы на компьютерах пользователя без запросов к серверу. Это экономит время и трафик, снижает нагрузку на сервер.

— Мощная инфраструктура (экосистема). Первые 10 лет этого не было и в помине. Затем количество готовых решений в открытом доступе так возросло, что работать с Javascript и его фреймворками стало удовольствием.

					МИВУ 09.03.04-8.000 ПЗ	Лист
						11
Изм.	Лист	№ докум.	Подпись	Дата		

— Простота и рациональность применения. Простую задачку можно решить за 5 минут, не надо делать лишнюю работу. Для сложных заданий есть варианты решения, можно подобрать лучший, адаптировать.

— Удобство пользовательских интерфейсов. Заполнение форм, выбор действий, активация кнопок, проверки ввода, реагирование на наведение / клики мыши и т.п. Это даёт потрясающий уровень юзабилити.

— Лёгкость освоения. Возможно, это дело привычки. Сначала код может показаться сложным, но к его синтаксису, логике быстро привыкаешь. Отображение действий ощутимо добавляет энтузиазма.

Как любой язык программирования, Javascript имеет некоторые недостатки (ограничения).

— Нет возможности чтения и загрузки файлов. Это ограничение функциональности на стороне клиента. Главная причина – соображения безопасности.

— Нестрогая типизация и вольная трактовка. Язык игнорирует явные нестыковки. Имеет место разная интерпретация данных. Нет возможности раннего выявления ошибок. Все недочёты выявляются уже на этапе работы.

— Нет поддержки удалённого доступа. Поэтому язык нельзя использовать для сетевых приложений. За это Javascript даже не считают полноценным языком программирования.

— Доступность для злоумышленников. В свободный скриптовый язык проще всего встроить фрагмент вредоносного кода, который может навредить пользователю. Надежда только на антивирус и фаервол.

Надо отметить, что сообщество пользователей Javascript активно улучшает язык, устраняя многие недостатки. Узких мест становится всё меньше. Браузеры постоянно совершенствуют работу с JS. Выход HTML 5 дал новый толчок к расширению возможностей скриптов.

3. Выбор и обоснование архитектуры системы (OpenServer)

Open Server — это портативный локальный WAMP/WNMP сервер, имеющий multifunctional управляющую программу и большой выбор подключаемых компонентов. Представленный пакет программ не является очередной любительской сборкой собранной «на коленке», это первый полноценный профессиональный инструмент, созданный специально для веб-разработчиков с учётом их рекомендаций и пожеланий.

Open Server представляет из себя так называемый WAMP комплекс, суть которого можно понять из расшифровки этой аббревиатуры:

1. Windows — операционная система, для работы в которой предназначен данный локальный сервер
2. Apache — web-сервер, который «поднимается» при запуске программы Open Server
3. MySQL — очень популярная система управления базами данных, которая является обязательным условием для работы многих движков сайтов, в том числе таких популярный, как Joomla и WordPress
4. PHP — интерпретатор серверного языка программирования, на котором написано большинство CMS и создано множество других веб-приложений

Программный комплекс имеет богатый набор серверного программного обеспечения, удобный, multifunctional продуманный интерфейс, обладает мощными возможностями по администрированию и настройке компонентов. Платформа широко используется с целью разработки, отладки и тестирования веб-проектов, а так же для предоставления веб-сервисов в локальных сетях. Имеется возможность настройки DNS сервера, запуска и остановки различных скриптов, тестирования доступности сайтов при разных скоростях интернета. Присутствует поддержка алиасов (дополнительных доменных имен одного и того же веб-сайта). Доступна работа в ограниченном режиме, когда нельзя

					МИВУ 09.03.04-8.000 ПЗ	Лист
						13
Изм.	Лист	№ докум.	Подпись	Дата		

вносить изменения в файл Hosts из-за проблем с правами администратора или блокировки файервола. Можно переносить сайты с движка Joomla на Wordpress и наоборот.

В общем, примечательным отличием Open Server является вариативность его инструментов. Он предлагает сразу два HTTP-модуля, три СУБД и множество PHP-модулей. При этом, переключение между ними не вызывает трудностей у оператора.

Также, несмотря на свою «увесистость», OpenServer остается достаточно быстрым. Open Server требует около четырех гигабайтов памяти на диске и 200 мегабайт RAM. Для современных компьютеров и физических носителей это не является слишком большим объемом. Тем более, что большой вес сполна компенсируется портативностью. OpenServer не засоряет системные файлы. Можно просто носить свой «сервер» на любом физическом накопителе и запускать на любом подходящем девайсе.

Большим преимуществом, которое высоко ценят новички, является простота установки и то, что OpenServer является бесплатным продуктом. Большинство системных файлов, которые необходимы для корректной работы операционной системы, установлены либо в самой системе, либо прикреплены к установщику. Если последний не находит на машине пользователя нужного элемента, он самостоятельно доустанавливает его. Такой подход сегодня используют большинство других программ.

Эта программа имеет огромный функционал, проста в использовании, обладает портативностью, а также является бесплатной. Все эти особенности дают понять, что OpenServer является хорошим выбором для создания веб-приложения на локальном сервере.

					МИВУ 09.03.04-8.000 ПЗ	Лист
						14
Изм.	Лист	№ докум.	Подпись	Дата		

4. Выбор и обоснование алгоритмов

4.1 Алгоритм работы с cookie файлами

Cookies - это механизм хранения данных браузером удаленного компьютера для идентификации возвращающихся посетителей и хранения параметров веб-страниц (например, переменных). Cookie файлы применяются для сохранения данных на стороне пользователя, на практике обычно используется для:

- аутентификации пользователя;
- хранения персональных предпочтений и настроек пользователя;
- отслеживания состояния сеанса доступа пользователя;
- сведения статистики о пользователях.

В разрабатываемом проекте cookies используются для временного хранения имени и id пользователя.

Алгоритм данной задачи выглядит следующим образом:

1. Проверяется наличие пользователя в БД.
2. Создаётся cookie на определённое время с использованием функция SetCookie().
3. В течение установленного времени авторизация пользователю не требуется.

4.2 Алгоритм обработки форм

Для передачи данных от пользователя к серверу в языке HTML есть специальная конструкция – формы. Формы предназначены для того, чтобы получать от пользователя информацию.

Данные, записанные в форму, отправляются для обработки специальной программе (например, скрипту на PHP) на сервере. В зависимости от введенных пользователем данных эта программа может

					МИВУ 09.03.04-8.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		15

формировать различные web-страницы, отправлять запросы к базе данных, запускать различные приложения и т.п.

Для создания формы в языке HTML используется тег FORM. Внутри него находится одна или несколько команд INPUT. С помощью атрибутов action и method тега FORM задаются имя программы, которая будет обрабатывать данные формы, и метод запроса, соответственно. Команда INPUT определяет тип и различные характеристики запрашиваемой информации. Отправка данных формы происходит после нажатия кнопки input типа submit.

Метод сообщает о цели запроса клиента. Протокол HTTP поддерживает достаточно много методов, но реально используются только три: POST, GET и HEAD. Основное преимущество POST запросов – это их большая безопасность и функциональность по сравнению с GET-запросами. Поэтому метод POST чаще используют для передачи важной информации, а также информации большого объема.

Алгоритм для работы с формой:

1. Показать форму (проверить авторизован ли пользователь – не для всех форм).
2. Для каждого поля проверить, что оно является заполненным (если данное поле обязательно к заполнению).
3. Если поле не заполнено, показать ошибку и не совершать отправку формы.
4. Если передаваемые поля имеют значения, то выполнить необходимые действия по обработке этих полей.

4.3 Алгоритм работы с БД

Алгоритм подключения к БД и написания запроса:

1. Первое, что нужно сделать, - это установить соединение с базой данных. Для этого воспользуемся функцией mysql_connect(), в которую в

					МИВУ 09.03.04-8.000 ПЗ	Лист
						16
Изм.	Лист	№ докум.	Подпись	Дата		

качестве параметров передадим `server = 'localhost'`, логин пользователя, пароль пользователя, имя используемой базы данных. Данная функция устанавливает соединение с сервером MySQL и возвращает указатель на это соединение или `FALSE` в случае неудачи.

2. Проверить успешность соединения.
3. С помощью функции `mysqli_query()` выполнить запрос к БД.
4. Проверить успешность выполнения запроса.

Вывод данных из таблицы в БД:

1. Подключиться к БД и выполнить функцию `mysqli_query()`.
2. С помощью цикла `while` и функции `mysqli_fetch_assoc()` обойти массив, который был получен из функции `mysqli_query()`, при этом выводя данные массива в нужном формате.

5. Описание программы

5.1 Описание сайта

Переходы между страницами осуществлялись через соответствующие ссылки в коде. Логика работы с базами данных была описана выше.

При открытии сайта пользователь попадает на главную страницу (Рисунок 1). На этой странице есть небольшая информация о фотографe, а также меню перехода на другие страницы сайта – блог, отзывы, контакты, вход (авторизация).

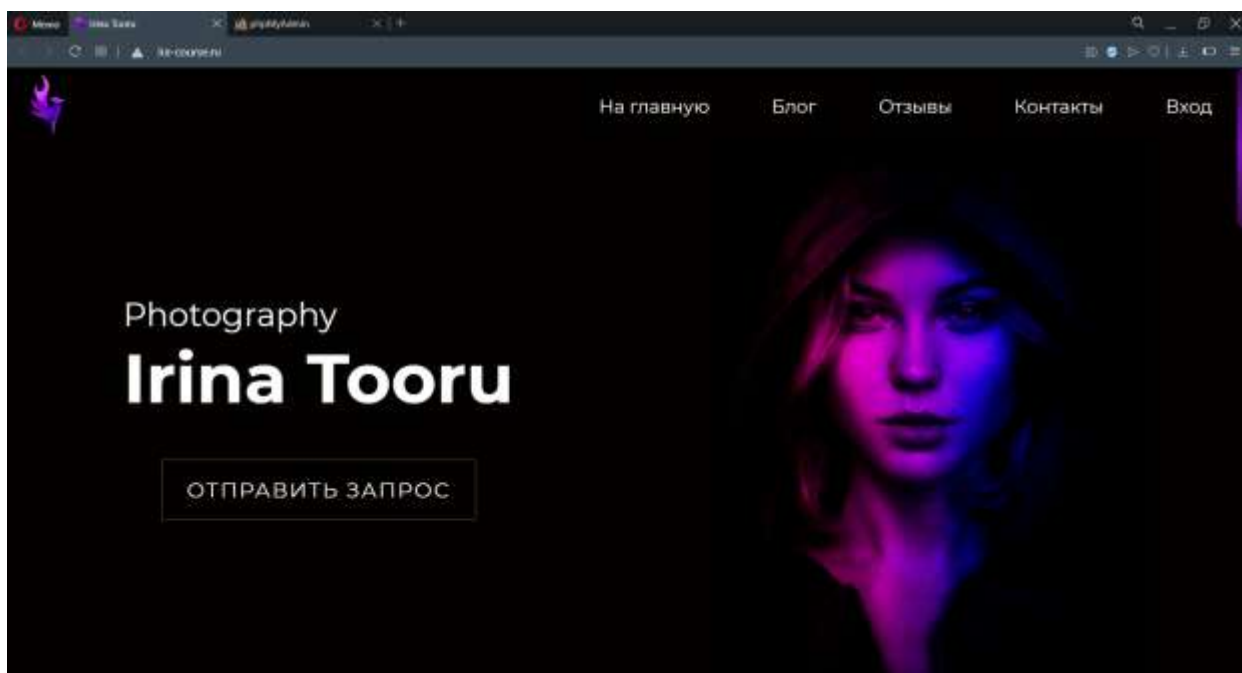


Рисунок 1 – скриншот главной страницы

На рисунке 2 изображен блог. Каждая статья блога представлена в виде отдельного блока (основное изображение, заголовок и дата, когда была написана эта статья) и при нажатии на определенную статью открывается полная информация о ней (рисунок 3) и комментарии к данной статье, которые могут оставлять только авторизованные пользователи (рисунок 4 и 5).

					МИВУ 09.03.04-8.000 ПЗ	Лист
						18
Изм.	Лист	№ докум.	Подпись	Дата		

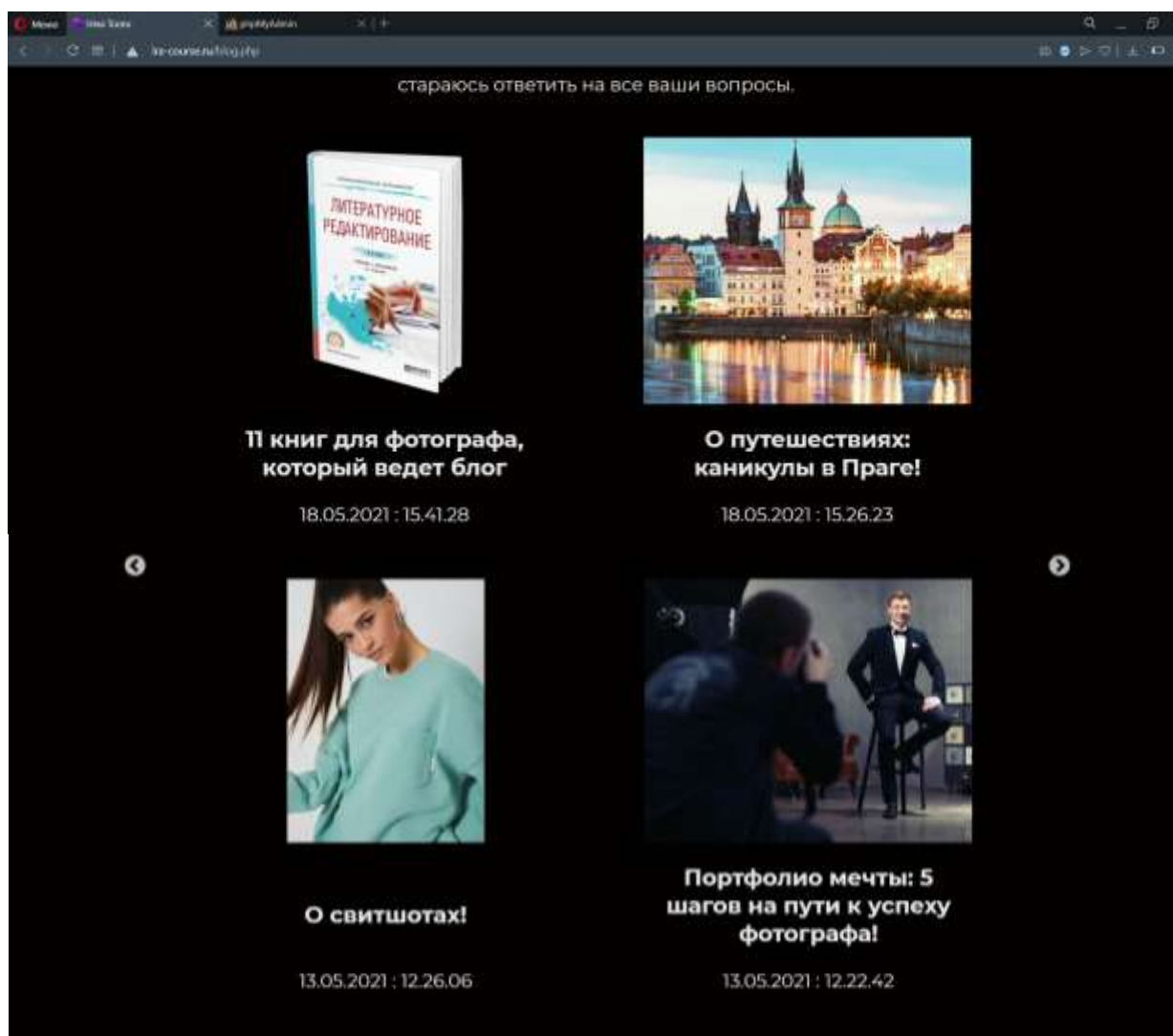


Рисунок 2 – скриншот страницы блога

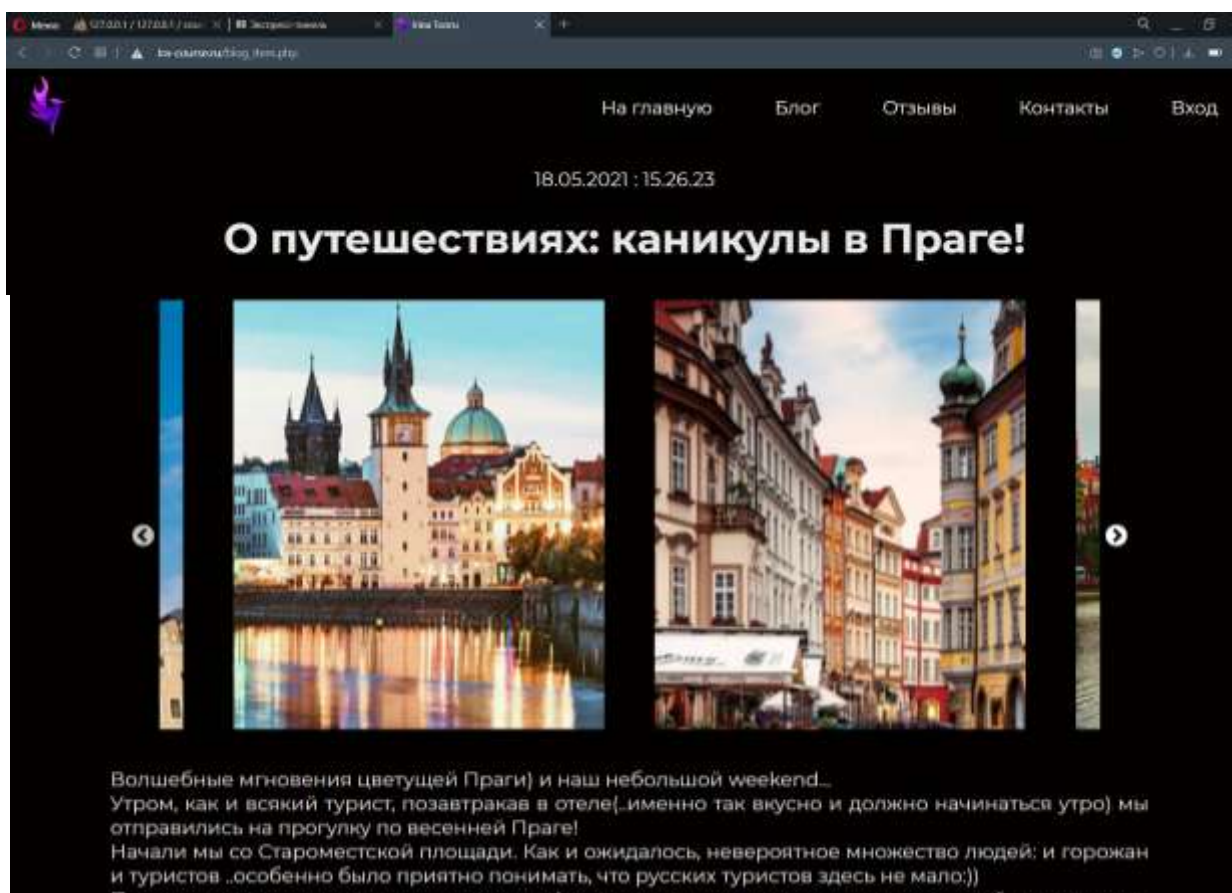


Рисунок 3 – статьи блога

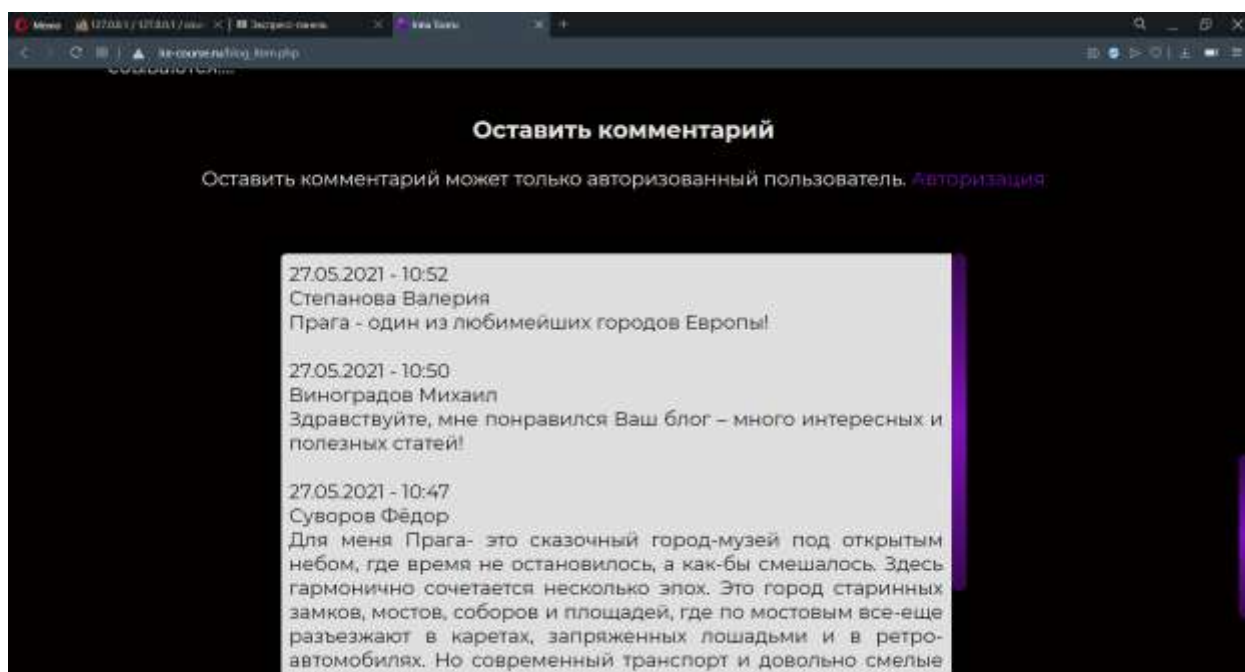


Рисунок 4 – комментарии к статье до авторизации

					МИВУ 09.03.04-8.000 ПЗ	Лист
						20
Изм.	Лист	№ докум.	Подпись	Дата		

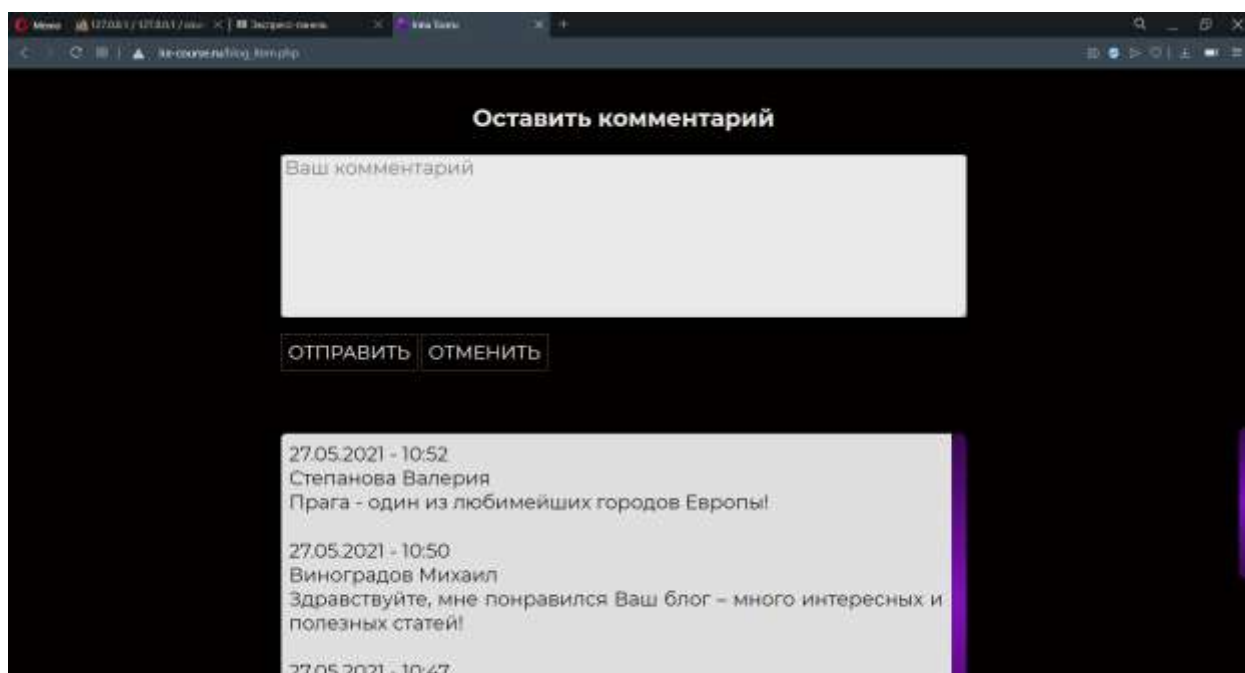


Рисунок 5 – комментарии к статье после авторизации

На рисунках 6 и 7 изображены отзывы. Авторизованный пользователь может отставить отзыв, а неавторизованный пользователь может только прочитать отзывы других пользователей.

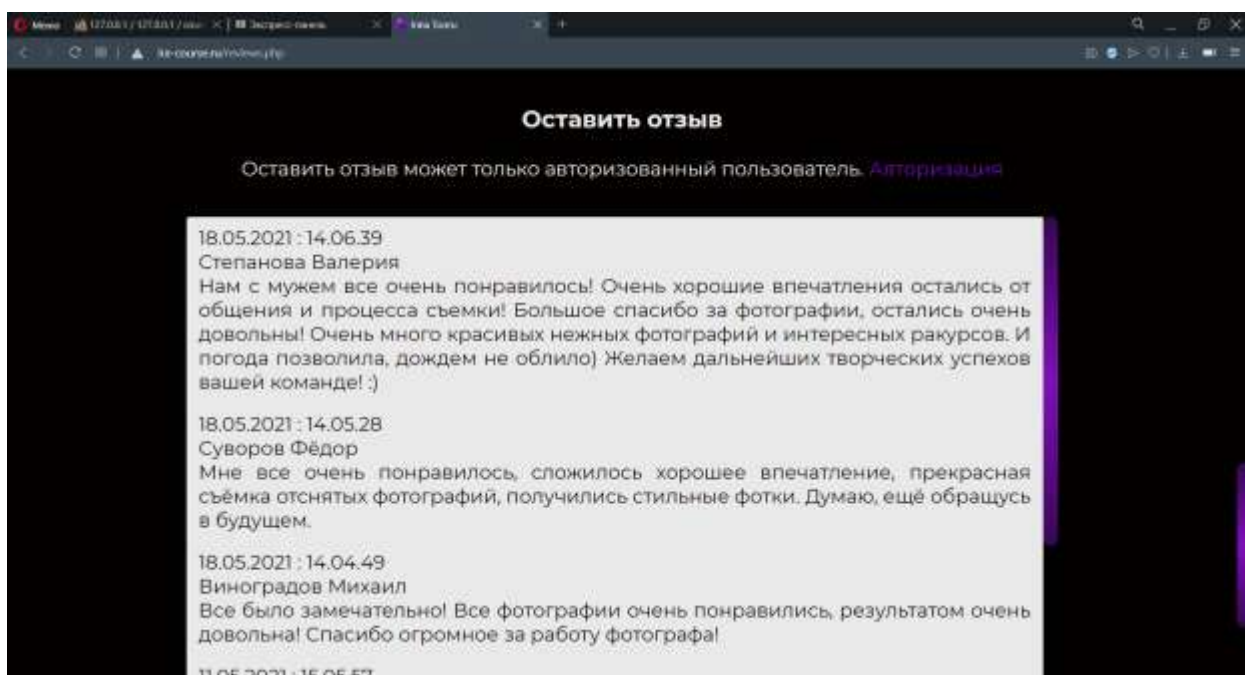


Рисунок 6 – отзывы до авторизации

Оставить отзыв

Ваш отзыв

18.05.2021 : 14.06.39
Степанова Валерия
Нам с мужем все очень понравилось! Очень хорошие впечатления остались от общения и процесса съемки! Большое спасибо за фотографии, остались очень довольны! Очень много красивых нежных фотографий и интересных ракурсов. И погода позволила, дождем не облило) Желаем дальнейших творческих успехов вашей команде! :)

18.05.2021 : 14.05.28
Суворов Фёдор
Мне все очень понравилось, сложилось хорошее впечатление, прекрасная съемка отснятых фотографий, получились стильные фотки. Думаю, ещё обращусь в будущем.

Рисунок 7 – отзывы после авторизации

Любой пользователь сайта может отправить сообщение, для этого он может заполнить форму, показанную на рисунке 8.

Форма для заполнения

Имя

Пол

☒ М ☐ Ж

Контактные данные

Тема сообщения

Сообщение

Рисунок 8 – форма для отправки сообщения

На рисунках 9 и 10 представлены формы авторизации и регистрации.

Рисунок 9 – форма авторизации

Рисунок 10 – форма регистрации

					МИВУ 09.03.04-8.000 ПЗ	Лист
						23
Изм.	Лист	№ докум.	Подпись	Дата		

5.2 Описание админ-панели

После успешной авторизации за администратора будет отображена админ-панель, где можно работать со всей информацией, которая содержится в базе данных. Администратор может добавить статью блога (рисунок 11).

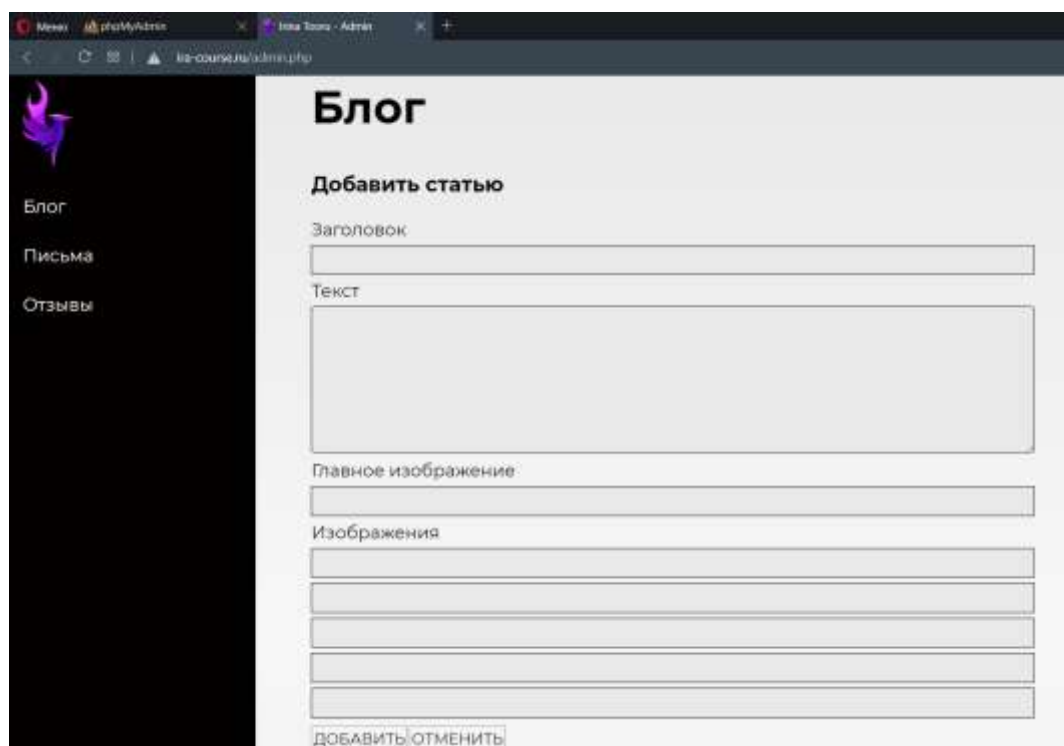
The image shows a web browser window with the URL 'http://localhost/admin.php'. The page title is 'Блог' (Blog). On the left is a dark sidebar with a purple flame logo and three menu items: 'Блог', 'Письма', and 'Отзывы'. The main content area is titled 'Добавить статью' (Add article). It contains several input fields: 'Заголовок' (Title), 'Текст' (Text), 'Главное изображение' (Main image), and 'Изображения' (Images). The 'Изображения' field has five sub-inputs. At the bottom are two buttons: 'ДОБАВИТЬ' (Add) and 'ОТМЕНИТЬ' (Cancel).

Рисунок 11 – форма добавления блога

В таблице на рисунке 12 представлен вывод статей блога в таблице. При нажатии на заголовок статьи, можно перейти в режим редактирования данной статьи (рисунок 13). Также можно удалить статью, при этом одновременно удаляются все комментарии к данной статье.

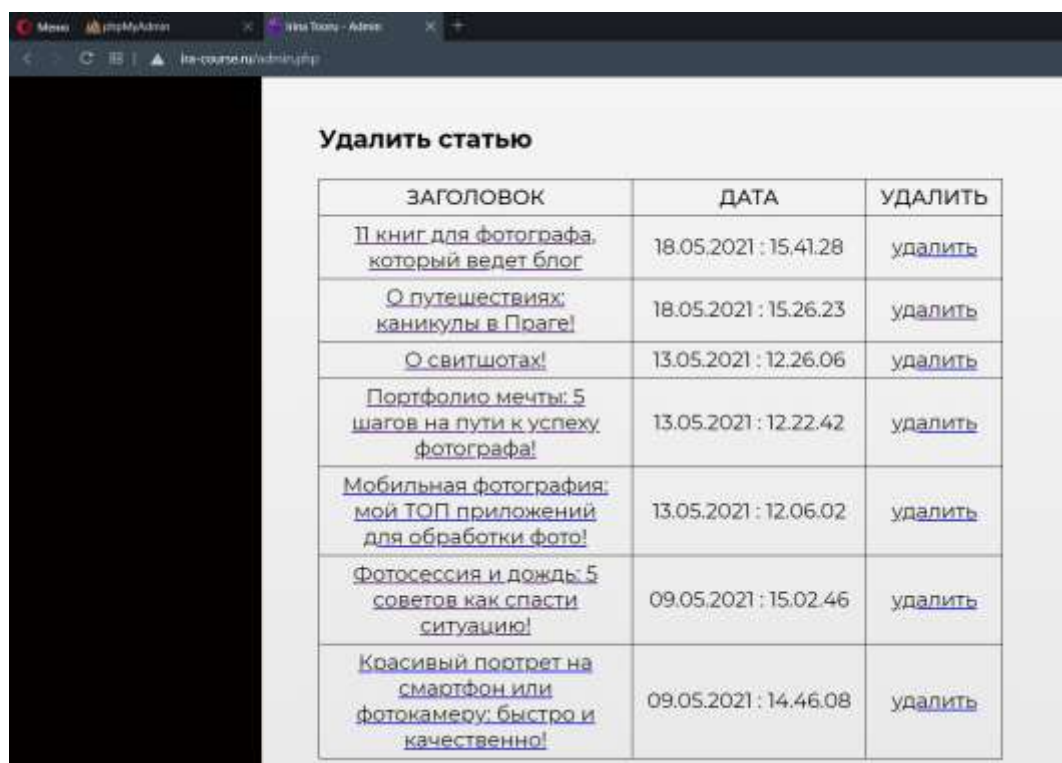


Рисунок 12 – вывод статей блога с возможностью удаления и перехода на изменение новости

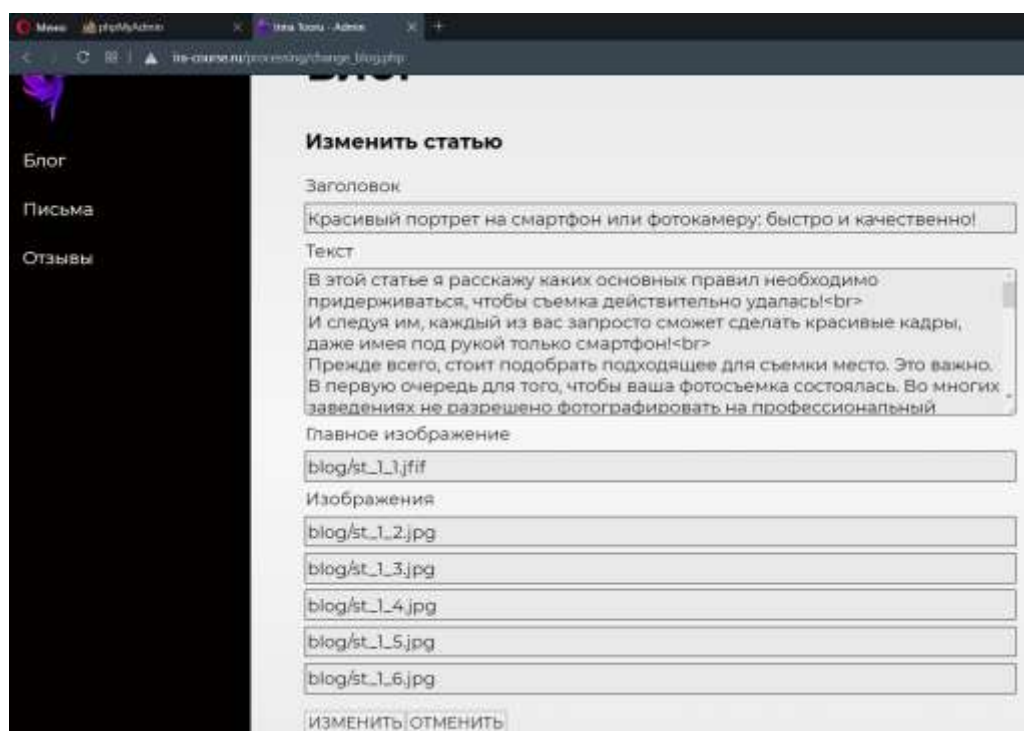


Рисунок 13 – изменение статьи блога

При выборе изменения статьи, перед администратором также появляются все комментарии к этой статье, которые он может удалить (рисунок 14).

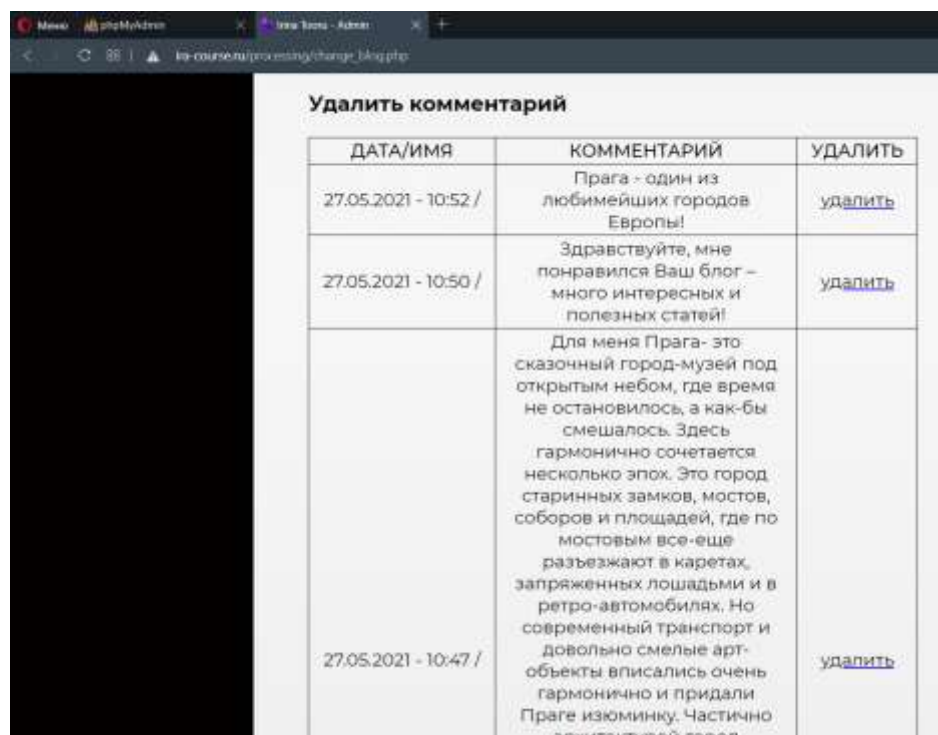


Рисунок 14 – вывод комментариев к статьям с возможностью удаления

Также администратор может просмотреть отзывы и удалить выбранный (рисунок 15).

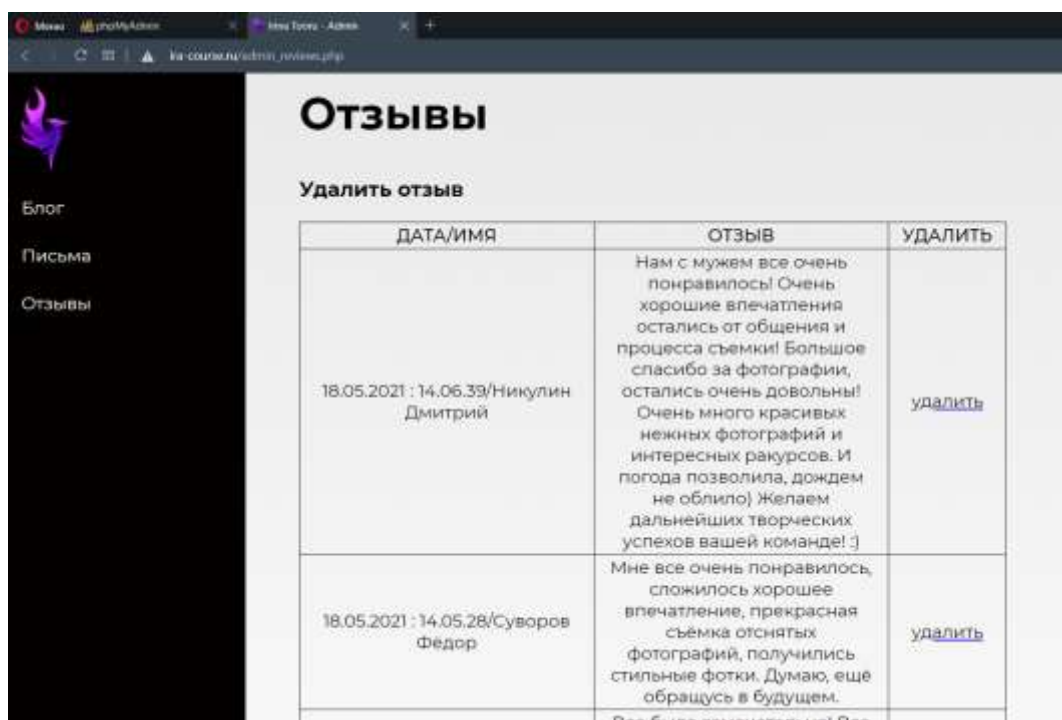


Рисунок 15 – вывод отзывов с возможностью удаления

5.3 Структура базы данных

В базе данных содержится пять таблиц. Таблицы «users», «reviews», «comments» связаны между собой по id пользователя, в них хранятся пользователи, отзывы и комментарии к блогу. Таблица «comments» связана с таблицей «blog» по id блога. Таблица «blog» хранит в себе информацию о статьях блога. Таблица «message» никак не связана с другими таблицами и хранит информацию о сообщениях с сайта. Структура базы данных представлена на рисунке 16.

Таблица «users». Первичный ключ id является автоинкрементным полем. В этой таблице содержится имя, логин и пароль пользователя.

Таблица «reviews». Первичный ключ id является автоинкрементным полем. В этой таблице содержится id пользователя, текст и дата написания отзыва.

Таблица «blog». Первичный ключ id является автоинкрементным полем. В этой таблице содержится id пользователя, текст и дата написания отзыва.

Таблица «comments». Первичный ключ id является автоинкрементным полем. В этой таблице содержится id блога, id пользователя, комментарий и дата написания комментария.

Таблица «message». Первичный ключ id является автоинкрементным полем. В этой таблице содержится имя, фамилия, пол, телефон, email. Тема сообщения, сообщение, дата написания сообщения.

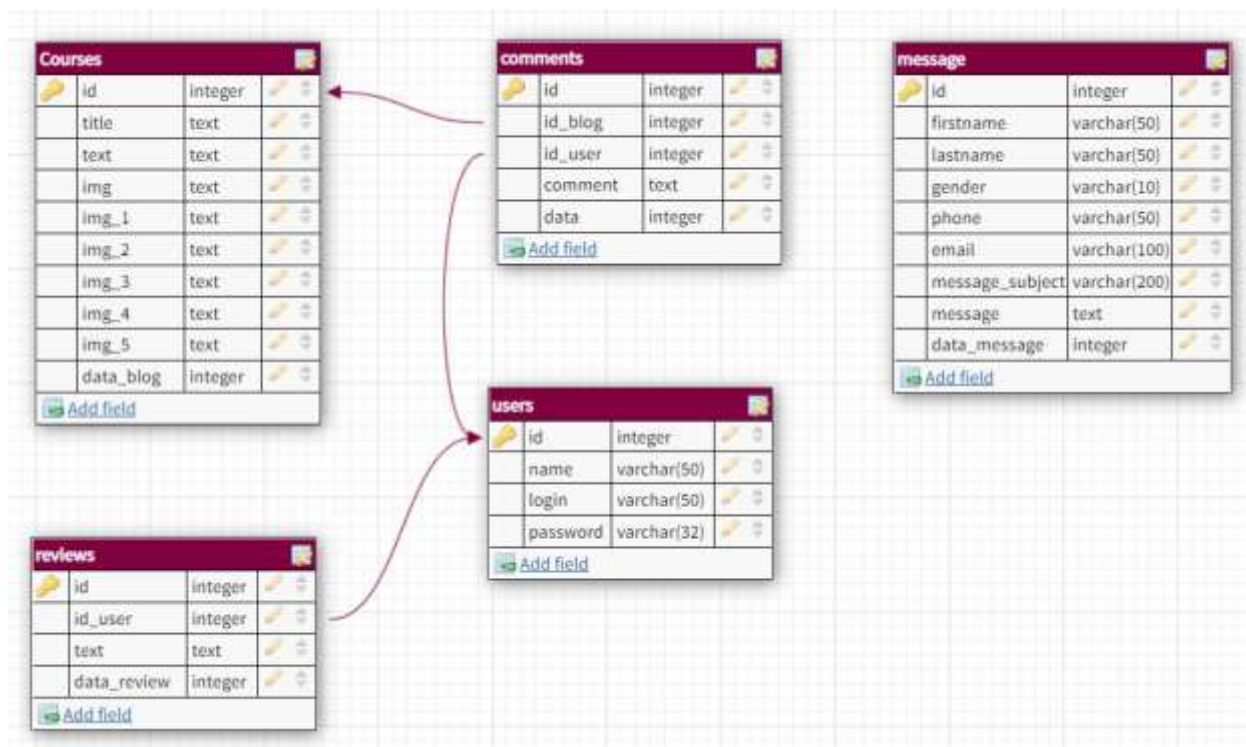


Рисунок 16 – Структура базы данных

Заключение

Результатом курсовой работы стал готовый к работе сайт фотографа. В ходе работы над ним был произведен анализ предметной области, создана база данных с необходимыми таблицами. Разработанный сайт обладает следующим функционалом:

- навигация по страницам сайта;
- возможность оставлять отзывы;
- возможность прочитать блог и оставить комментарий к статье блога;
- возможность написать сообщение фотографу;
- для администратора есть возможности добавления / изменения / удаления статей блога, просмотра / удаления комментариев к статьям блога, отзывов и писем.

Для реализации сайта были использованы следующие средства:

- редактор исходного кода Sublime Text 3;
- локальный сервер Open Server;
- база данных MySQL;
- язык гипертекстовой разметки HTML и каскадные таблицы стилей CSS;
- языки программирования PHP и JavaScript.

Таким образом, в курсовой работе были реализованы все пункты технического задания.

Список используемой литературы

1. Чак М., Кеннеди Б. HTML и XHTML. Подробное руководство. 6-е издание. - Символ-Плюс, 2008 г. - 748с.:
2. Дакетт Д. Основы веб-программирования с использованием HTML, XHTML и CSS. 2-е издание. - Эксмо, 2010 г. - 768с.
3. Веллинг Л., Томсон Л. Разработка веб-приложений с помощью PHP и MySQL. - Вильямс 2010г. - 880с.

					МИВУ 09.03.04-8.000 ПЗ	Лист
						30
Изм.	Лист	№ докум.	Подпись	Дата		

Приложение

Приложение 1. Программный код

При авторизации пользователя из массива `$_POST` мы получаем данные с формы - логин и пароль. Осуществляем проверку на то, является ли пользователь администратором, если да, то переходим на страницу администратора, иначе пробуем найти его в БД пользователей. При успешном нахождении устанавливаем cookie.

<?

```
$login = $_POST['login'];
```

```
$pass = $_POST['pass'];
```

```
$pass = md5($pass);
```

```
if($login == 'admin' && $pass == md5('admin'))
```

```
{
```

```
    header('Location: /admin.php');
```

```
    exit();
```

```
}
```

```
$db = mysqli_connect("localhost", "root", "root", 'course');
```

```
$sql = mysqli_query($db, "SELECT * FROM `users` WHERE `login` = '$login' and `password`  
= '$pass'");
```

```
$count = 0;
```

```
while ($result = mysqli_fetch_array($sql)) {
```

```
    $count++;
```

```
    setcookie('user', $result['name'], time() + 3600, "/");
```

```
    setcookie('user_id', $result['id'], time() + 3600, "/");
```

```
}
```

```
if($count == 0){
```

```
    echo "Такой пользователь не найден";
```

```
    exit();
```

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | МИВУ 09.03.04-8.000 ПЗ | Лист |
| | | | | | | 31 |
| Изм. | Лист | № докум. | Подпись | Дата | | |


```

    }

    $db->close();

    header('Location: /');
?>

```

Использование cookie. Если cookie установлена, то пользователь может написать отзыв, иначе его попросят авторизоваться.

```

<? if($_COOKIE['user'] != ""): ?>
    <textarea name="text" class="textarea_review_user" placeholder="Ваш отзыв"
required></textarea><br>
    <input type="submit" value="Отправить" class="button_contacts">
    <input type="reset" value="Отменить" class="button_contacts">
<? else: ?>
    <p>Оставить отзыв может только авторизованный пользователь. <a href="entance.php"
class="a_purple">Авторизация</a></p>
<? endif; ?>

```

Добавление новых записей в таблицу базы данных происходило с использованием формы. Методом POST эти файл отправлялись в php код и добавлялись в базу данных. Ниже представлен код добавления новой записи в таблицу «blog».

```

<form method="POST">
    <h2>Добавить статью</h2>

    <p>Заголовок</p><br>
    <input type="text" name="title" class="text_input" required><br>
    <br><p>Текст</p><br>
    <textarea name="text" class="big_text_input" required></textarea><br>

    <br><p>Главное изображение</p><br>
    <input type="text" name="img" class="text_input" required><br>
    <br><p>Изображения</p><br>
    <input type="text" name="img_1" class="text_input" required><br><br>

```

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | МИВУ 09.03.04-8.000 ПЗ | Лист |
| | | | | | | 32 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

```

<input type="text" name="img_2" class="text_input" required><br><br>
<input type="text" name="img_3" class="text_input"><br><br>
<input type="text" name="img_4" class="text_input"><br><br>
<input type="text" name="img_5" class="text_input"><br><br>

<input type=submit value="Добавить" class="button_form">
<input type=reset value="Отменить" class="button_form">

<?php
if(!empty($_REQUEST)){
    if($_REQUEST['title'] != null)
    {
        // Переменные с формы
        $title = $_REQUEST['title'];
        $text = $_REQUEST['text'];
        $img = $_REQUEST['img'];
        $img_1 = $_REQUEST['img_1'];
        $img_2 = $_REQUEST['img_2'];
        $img_3 = $_REQUEST['img_3'];
        $img_4 = $_REQUEST['img_4'];
        $img_5 = $_REQUEST['img_5'];

        try {
            $db = mysqli_connect("localhost", "root", "root",
'course');

            if ($db == false){
                print("Ошибка: Невозможно подключиться к MySQL " .
mysqli_connect_error());
            }

            $sql = mysqli_query($db, "INSERT INTO `blog` (`title`,
`text`, `img`, `img_1`, `img_2`, `img_3`, `img_4`, `img_5`, `data_blog`) VALUES
('$title', '$text', '$img', '$img_1', '$img_2', '$img_3', '$img_4', '$img_5',
'".time()."')");

            if (!$sql) {
                echo '<p>Произошла ошибка: ' . mysqli_error($db) .
'</p>';
            }

            $db -> close();

```

| | | | | | | |
|------|------|----------|---------|------|------------------------|------|
| | | | | | МИВУ 09.03.04-8.000 ПЗ | Лист |
| | | | | | | 33 |
| Изм. | Лист | № докум. | Подпись | Дата | | |

```

    }
    catch (PDOException $e) {
        print "Ошибка!: " . $e->getMessage() . "<br/>";
    }
}
}
?>

```

</form>

При изменении данных в файл обработчик через ссылку передавался id этого элемента:

```
<a href="processing/change_blog.php?ID=<?=$Id?>">
```

После чего в другом файле методом GET этот id принимается:

```
<? $ID = $_GET['ID'];?>
```

По полученному id в базе данных находится запись, после чего форма изменения заполняется найденными значениями:

```

<form action="change.php" method="POST">
    <h2>Изменить статью</h2>

    <?
    $db = mysqli_connect("localhost", "root", "root", 'course');
    $sql = mysqli_query($db, "SELECT * FROM `blog` WHERE `Id` = {$ID}");

    $result = mysqli_fetch_array($sql);
    $title = $result['title'];
    $text = $result['text'];
    $img = $result['img'];
    $img_1 = $result['img_1'];
    $img_2 = $result['img_2'];
    $img_3 = $result['img_3'];
    $img_4 = $result['img_4'];
    $img_5 = $result['img_5'];

```

					МИВУ 09.03.04-8.000 ПЗ	Лист
						34
Изм.	Лист	№ докум.	Подпись	Дата		

```

$db -> close();
?>
<p>Заголовок</p><br>
<input type="text" name="title" class="text_input" value="<?=$title?>"
required><br>
<br><p>Текст</p><br>
<textarea name="text" class="big_text_input"
required><?=$text?></textarea><br>

<br><p>Главное изображение</p><br>
<input type="text" name="img" class="text_input" value="<?=$img?>"
required><br>
<br><p>Изображения</p><br>
<input type="text" name="img_1" class="text_input" value="<?=$img_1?>"
required><br><br>
<input type="text" name="img_2" class="text_input" value="<?=$img_2?>"
required><br><br>
<input type="text" name="img_3" class="text_input"
value="<?=$img_3?>"><br><br>
<input type="text" name="img_4" class="text_input"
value="<?=$img_4?>"><br><br>
<input type="text" name="img_5" class="text_input"
value="<?=$img_5?>"><br><br>

<input type="text" name="id" class="text_input" value="<?=$ID?>"
style="display: none"><br>

<input type="submit" value="Изменить" class="button_form">
<input type="reset" value="Отменить" class="button_form">
</form>

```

Обработка изменения формы.

```

<?php
if(!empty($_REQUEST)){
    if($_REQUEST['title'] != null)
    {
        // Переменные с формы
        $title = $_REQUEST['title'];
        $text = $_REQUEST['text'];

```

					МИВУ 09.03.04-8.000 ПЗ	Лист
						35
Изм.	Лист	№ докум.	Подпись	Дата		

```

$img = $_REQUEST['img'];
$img_1 = $_REQUEST['img_1'];
$img_2 = $_REQUEST['img_2'];
$img_3 = $_REQUEST['img_3'];
$img_4 = $_REQUEST['img_4'];
$img_5 = $_REQUEST['img_5'];
$ID = $_REQUEST['id'];

if(!empty($title)){
    $db = mysqli_connect("localhost", "root", "root", 'course');
    $sql = mysqli_query($db, "UPDATE `blog` SET `title` =
'$title',`text` = '$text',`img` = '$img',`img_1` = '$img_1',`img_2` =
'$img_2',`img_3` = '$img_3',`img_4` = '$img_4',`img_5` = '$img_5' WHERE `Id` =
'$ID'");

    if (!$sql) {
        echo "<p>Произошла ошибка: " . mysqli_error($db) . "</p>";
    }
    $db -> close();
}
}
header('Location: /admin.php');
}
?>

```

При удалении записи из таблицы сначала в ссылке передается id элемента, который нужно удалить, после чего происходит удаление:

```

<?
$db = mysqli_connect("localhost", "root", "root", 'course');
$sql = mysqli_query($db, "DELETE FROM `blog` WHERE `Id` = {@$_GET['ID']}");
$sql_1 = mysqli_query($db, "DELETE FROM `comments` WHERE `id_blog` = {@$_GET['ID']}");
if ($sql) {
    header('Location: /admin.php');
}
else {
    echo '<p>Произошла ошибка: ' . mysqli_error($db) . '</p>';
}
$db -> close();
?>

```

В данной курсовой работе есть несколько слайдеров. Они реализованы с помощью Slick.

Код:

```
$('.slider').slick({
  centerMode: true,
  centerPadding: '60px',
  slidesToShow: 2,
  autoplay: true,
  speed: 1000,
  autoplaySpeed: 2000,
  responsive: [
    {
      breakpoint: 768,
      settings: {
        centerMode: true,
        slidesToShow: 1,
        centerPadding: '30px',
      }
    },
  ],
});
```

					МИВУ 09.03.04-8.000 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		37