МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра теоретических основ компьютерной безопасности и криптографии

Система проектирования и поддержки защищенного сайта

АВТОРЕФЕРАТ
дипломной работы

студентки 6 курса 631 группы
специальности 10.05.01 Компьютерная безопасность
факультета компьютерных наук и информационных технологий
Селяниной Виктории Александровны

Научный руководитель
ассистент _____ А. А. Лобов
22.01.2022 г.

Заведующий кафедрой
д. ф.-м. н., доцент _____ М. Б. Абросимов
22.01.2022 г.

Саратов 2022

ВВЕДЕНИЕ

Внедрение средств вычислительной техники, доступность информации, объем и скорость её обработки являются важнейшими факторами развития науки, культуры и всех сфер жизнедеятельности человека. Информация и данные все чаще рассматриваются как жизненно важные ресурсы, которые должны быть организованы таким образом, чтобы ими можно было легко пользоваться.

Основные идеи современной информационной технологии основываются на том, что данные должны быть организованы в базы данных, с целью адекватного отображения изменяющегося реального мира и удовлетворения информационных потребностей пользователей.

Любая информационная система представляет собой программный комплекс, функции которого состоят в поддержке надежного хранения информации, выполнении для данного приложения преобразований информации и вычислений, предоставлении пользователям удобного и легко усваиваемого интерфейса.

С развитием и распространением сети Интернет информационные системы стали более интерактивными, масштабируемыми и доступными обычным пользователям.

Современный сайт – это полноценный программный продукт, который отслеживает действия пользователей, позволяет им между собой общаться и предлагает множество полезных сервисов в зависимости от поставленных владельцем сайта задач. На данный момент сайтов существует невероятное количество. Web-приложения имеют широкую распространенность.

Дипломная работа состоит из введения, 4 разделов, заключения, списка использованных источников и 2 приложений. Общий объем работы – 66 страниц, из них 45 страниц – основное содержание, включая 45 рисунков, список использованных источников из 16 наименований.

КРАТКОЕ СОДЕРЖАНИЕ

1 Что такое сайт?

Сайт представляет собой массив связанных данных, имеющий персональный для каждого сайта адрес, который принято называть URL. Каждый веб-сайт имеет своего владельца, им может быть как физическое, так и юридическое лицо. Веб-сайты в совокупности составляют Всемирную паутину. Для прямого доступа клиентов к сайтам на серверах был специально разработан протокол HTTP.

Страницы сайтов – это набор текстовых файлов, размеченных на языке HTML. Язык HTML позволяет форматировать текст, различать в нём функциональные элементы, создавать гипертекстовые ссылки и вставлять в отображаемую страницу изображения, звукозаписи и другие мультимедийные элементы. Отображение страницы можно изменить добавлением стилей на языке CSS, что позволяет централизовать в определённом файле все элементы форматирования (размер и цвет заглавных букв, размер и вид блока вставки) или сценариев на языке программирования JavaScript, с помощью которого имеется возможность просматривать страницы с событиями или действиями.

2 Протокол HTTP

Основой HTTP является технология «клиент-сервер», то есть предполагается существование:

- клиентов, которые инициируют соединение и посылают запрос;
- серверов, которые ожидают соединения для получения запроса, производят необходимые действия и возвращают обратно сообщение с результатом.

Задача, которая традиционно решается с помощью протокола HTTP – обмен данными между пользовательским приложением и веб-сервером. Обмен сообщениями идёт по обыкновенной схеме «запрос-ответ». Для поддержки авторизованного доступа в HTTP используются cookies.

3 Выбор веб сервера

Apache Web Server и Nginx – два самых широко распространенных веб-сервера. Оба решения способны работать с разнообразными рабочими нагрузками и взаимодействовать с другими приложениями. Каждый из них имеет собственные преимущества и важно понимать какой веб-сервер выбрать в различных ситуациях.

Система ориентирована на Nginx, так как он является асинхронным обратным прокси-сервером. Это означает, что он умеет кешировать некоторые ответы, чтобы снизить нагрузку на процессор и дисковую систему, и выполняет запросы асинхронно, чем достигается наименьшее время простоя. Работают два сайта. Один сайт находится в /var/www/html, предназначен для разработчика. Второй находится внутри папки files – это тот сайт, который разработчик разрабатывает. Сайт разработчика имеет доступ к папке files и может её редактировать. Компоненты системы представлены на рисунках 1 и 2 в виде дерева решения.

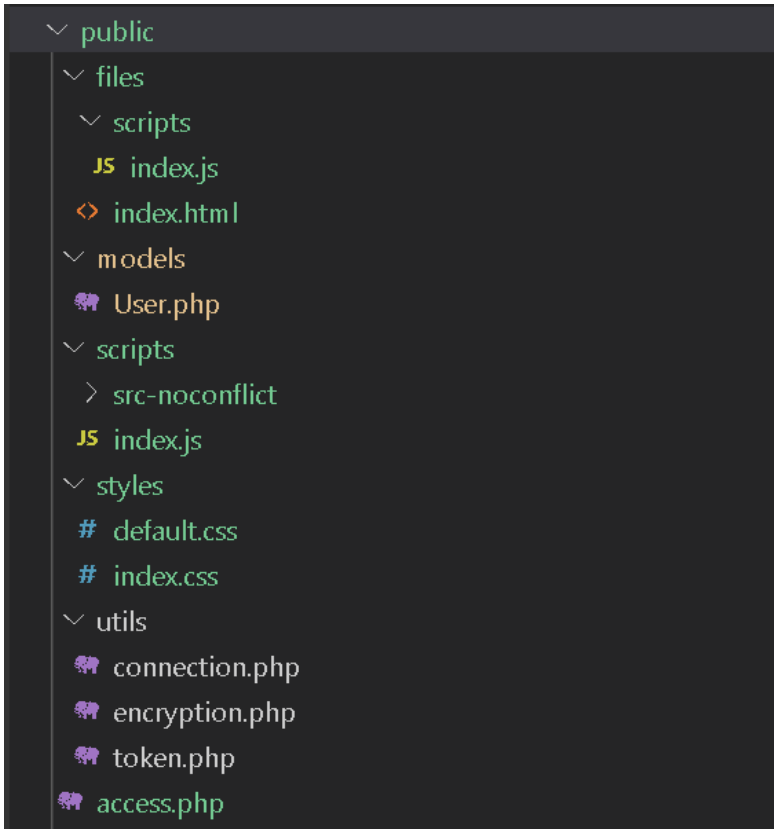

Рисунок 1 – Дерево решения

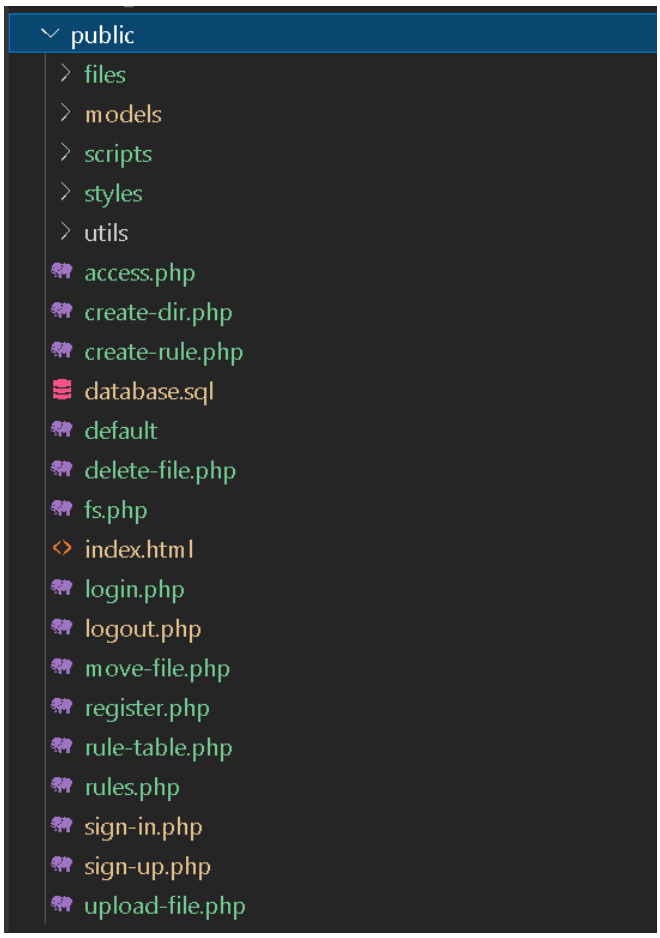

Рисунок 2 – Дерево решения

4 Задачи системы проектирования сайтов

Основными задачами проектирования сайтов являются:

- Идентификация, аутентификация и авторизация пользователя;
- Разграничение доступа;
- Управление ресурсами.

4.1 Идентификация, аутентификация и авторизация пользователя

Одной из важных задач обеспечения защиты от НСД является использование методов и средств, позволяющих одной (проверяющей) стороне убедиться в подлинности другой (проверяемой) стороны. Прежде чем получить доступ к ресурсам компьютерной системы, пользователь должен пройти процесс первичного взаимодействия с компьютерной системой, который включает идентификацию и аутентификацию.

Для того, чтобы пользователю не приходилось вводить одни и те же данные для сайтов, которые посещаются регулярно, используют cookie. Они имеют два флага: secure и httponly. Secure передаётся только по протоколу https. Второй флаг означает, что httponly cookie не доступны из JavaScript.Таким образом осуществляется защита от межсайтового скриптинга. В cookie устанавливается токен – это информация, которая выдается навсегда. В дальнейшем, если токен предоставлен, можно определить пользователя, которому он был выдан, и назначить ему соответствующие права.

В качестве базы данных была использована MariaDB. В базе данных хранятся 3 таблицы: users, roles и rules.

Таблица пользователей содержит 5 зарегистрированных пользователей и администратора (суперпользователь). При регистрации пользователей, файл sign-up.php добавляет новых пользователей в базу данных. При входе в систему sign-in.php формирует токен и флаг httponly со значением true. Таким образом осуществляется защита от межсайтового скриптинга. При выходе пользователя из системы с помощью файла logout.php удаляются cookie.

4.2 Разграничение доступа

Права доступа определяют набор действий, разрешённых для выполнения субъектам (например, пользователям системы) над объектами данных. Для этого требуется некая система для предоставления субъектам различных прав доступа к объектам. Это система разграничения доступа субъектов к объектам, которая рассматривается в качестве главного средства защиты от несанкционированного доступа к информации.

Разграничение доступа осуществляется с помощью правил, составляемых администратором. В правилах описывается кому (какой роли) что можно делать (метод) и с чем (ресурсом). В своём проекте я определила три роли:

- Guest (неавторизованный пользователь);
- User (авторизованный пользователь);

- Администратор (суперпользователь).

Гость имеет идентификатор 1, пользователь авторизованный идентификатор 2, а администратор идентификатор 3.

Введём в поисковой строке localhost/admin/login.php и осуществим вход в систему под пользователем admin. При открытии сайта разработчика появляется основная страница редактора, показано на рисунке 3. Слева находится панель навигации, в ней администратор может создавать и редактировать правила, а также осуществлять все действия по созданию, перемещению и удалению директорий и файлов. Справа располагается панель редактора, в ней отображается код файлов. Эта панель имеет следующие возможности: создание новых файлов с заданным путём, редактирование и сохранение файлов.

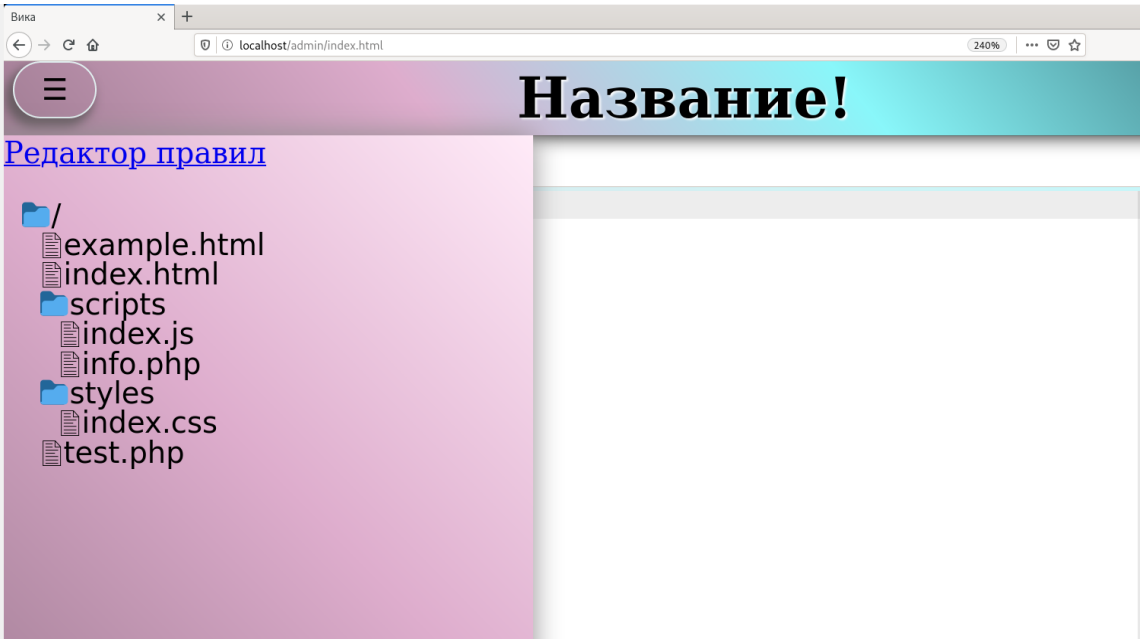

Рисунок 3 – Основная страница сайта разработчика

Для создания и редактирования правил администратору необходимо перейти по ссылке: «Редактор правил».

Для реализации правил может быть выбран один из двух принципов:

1) запрещено все, что явно не разрешено;
2) разрешено все, что явно не запрещено.

В данном проекте я использовала второй пункт.

Для разграничения действий с ресурсами были использованы следующие HTTP-методы:

- GET – получение ресурса;
- POST – создание ресурса;
- PUT – обновление ресурса;
- DELETE – удаление ресурса.

Администратор в редакторе правил может создавать новые и редактировать существующие правила. Для этого ему необходимо выбрать роль, для которой создается или редактируется правило, указать ресурс и выбрать методы, которые он хочет запретить. Создадим правила для ролей. Запретим доступ к сайту разработчика пользователю и гостю. Для проверки правила пользователю необходимо совершить вход на сайт разработчика. При попытке входа на страницу localhost/admin/index.html у пользователя появляется страница с кодом ошибки 403, что означает запрет доступа, показано на рисунке 4. Таким образом, доступ к сайту разработчика имеет только администратор.

Рисунок 4 – Запрет доступа для пользователя

4.3 Управление ресурсами

Ресурсы обычно хранятся в директориях на сервере. Под ресурсами будем понимать файлы. Управление ресурсами сайта подразумевает умение загружать, изменять, перемещать и удалять файлы.

Администратор внутри сайта разработчика создает, добавляет и редактирует файлы, а они отображаются внутри виртуальной папки files у пользователей. Это показано на рисунках 5-6.

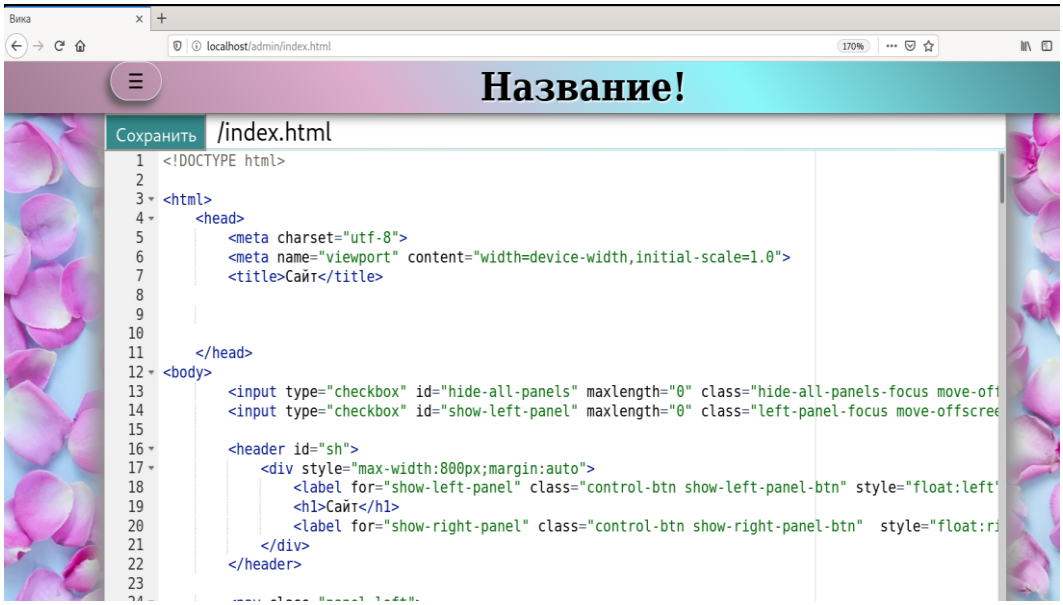

Рисунок 5 – Созданный администратором файл index.html

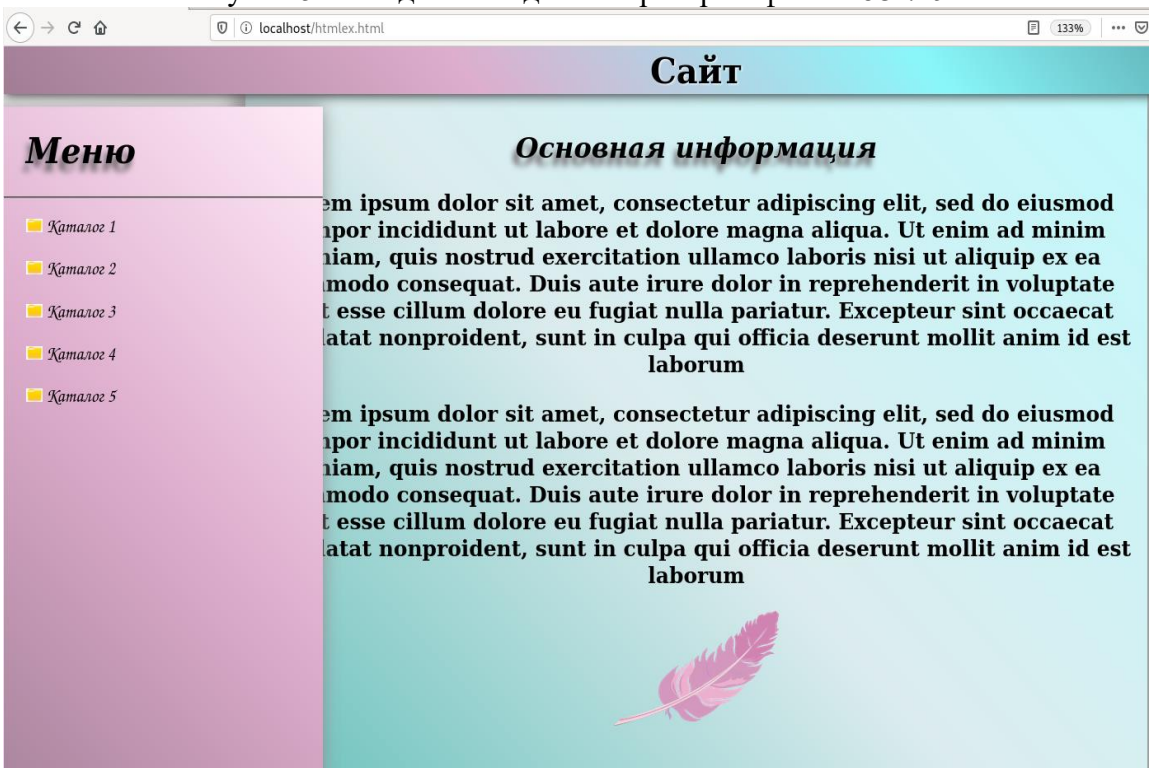

Рисунок 6 – Отображение файла index.html на сайте пользователей

Редактор сайта умеет распознавать различные языки программирования и выполнять соответствующие подсветки синтаксиса.

Администратор может создавать, удалять папки и файлы, перемещать файлы из одной директории в другую, загружать файлы в папки. Всё это продемонстрировано в дипломной работе.

ЗАКЛЮЧЕНИЕ

В ходе выполнения дипломной работы была разработана система проектирования сайтов, которая включает подсистему идентификации, аутентификации и авторизации, разграничения доступа и редактирования содержимого сайта. Были разработаны программы на языке программирования php, которые выполняют регистрацию и вход пользователей, программы для присвоения роли пользователям по идентификаторам и разграничение прав пользователей с помощью методов HTTP запросов. Для редактирования содержимого сайта был разработан редактор файлов на языке программирования JavaScript, предназначенный для создания, загрузки, изменения, перемещения и удаления файлов.