Содержание

Введение

Теоретическая часть

Принцип работы HTTP

Методы HTTP

Базы данных

Введенние

Актуальность данной темы состоит в том, что в настоящее время веб-приложения имеют стабильный спрос, так как они используются повсеместно и полностью вошли в нашу жизнь. Веб-приложения позволяют решить много задач, повысить эффективность работы, облегчить жизнь людям. Они могут быть использованы как в медицине, так и в образовании, и в повседневной деятельности людей. Также для создания веб-приложения требуется немало технических знаний и навыков, без которых сложно разобраться в работе сайтов.

Почему был выбран NodeJS? NodeJS – это кроссплатформенная среда выполнения кода JavaScript, которая позволяет разработчикам создавать серверы и приложения. Эта платформа используется для создания высоконагруженных веб-приложений, которые работают на стороне сервера. Кроме того, NodeJS имеют обширную документацию и множество готовых модулей, что сокращает время на создание приложения и облегчает их разработку. Всё это делает данную тему актуальной.

В рамках данного проекта я поставил перед собой цель: создать веб-приложение на NodeJS. Исходя из цели курсовой работы можно поставить перед собой следующий задачи:

* Изучение принципа работы протоколов HTTP и HTTPS
* Изучение методов HTTP
* Изучение базы данных MongoDB
* Написание веб-приложения
* Проектирование дизайна и пользовательского интерфейса

1. Теоретическая часть

Принцип работы протоколов HTTP и HTTPS

Протокол HTTP – это «протокол передачи гипертекста». Он используется для передачи данных и информации, которая происходит между стороной сервера и браузера (клиентом). Клиентское приложение формирует запрос и отправляет его на сервер, после чего сервер обрабатывает данный запрос, формирует ответ и передаёт его обратно клиенту. После этого клиентское приложение может продолжить отправлять другие запросы, которые будут обработаны тем же самым способом. Кроме того, именно протокол определяет правила передачи данных.

HTTP не имеет связи между двумя запросами, поэтому порой возникают проблемы для пользователя, который пытается взаимодействовать с корзиной на определенной странице последовательно, например, при использовании корзины с понравившимися товарами. На помощь приходит куки, позволяющие использовать сессии с сохранением состояния. Куки добавляются к потоку HTTP, при этом позволяя сессия делится необходимыми данными.

Так что же тогда такое HTTPS и в чем различия между ними? HTTPS – это расширение протокола HTTP, который использует шифрование данных. Как мы уже знаем, HTTP служит транспортом для передачи информации, при этом он никак не защищен. Защиту данных обеспечивает протокол SSL/TLS, который как раз и шифрует данные. Этот протокол хорош тем, что обеспечивает двум незнакомым участником сети защищенное соединение через незащищенный канал.

Представим, что мы хотим передать «секретный данные» другому человеку. Мы отправляем по почте ящик, который заперт замком, чтобы никто не смог украсть информацию. Ящик был доставлен до адресата, но он его не может открыть, так как у него нет ключа. Он вешает свой замок на этот ящик и отправляет его нам. Мы получаем ящик с двумя замками, снимаем свой и отправляем снова. Наконец, адресат получает ящик, на котором висит только его замок, и он может его открыть.

Приблизительно так состоит принцип работы у SSL/TLS. То есть компьютер и сервер сначала выбирают общий секретный ключ, а дальше обмениваются информацией, шифруя всё при помощи этого ключа. При каждой новой связи этот ключ генерируется заново. Однако для точной надежности не хватает одного, а именно гарантии того, что «наш собеседник» является тем, кем является.

Представим, что нашу посылку получил не адресат, а другой человек. Он может повесить свой замок, отправить нам, и мы отправляем ящик с одним замком совершенно не тому человеку.

В интернете избежать такую ситуацию помогает цифровой сертификат – электронный документ, который используется для идентификации сервера. Пользователю не нужно иметь сертификат, но сервер (сайт) должен его иметь. Сертификат подтверждает только 2 вещи, а именно: 1) Лицо, которому он выдан, существует. 2) Оно управляет сервером, который указан в сертификате. Иными словами, нашу посылку получает именно тот человек, который нужен. Сертификаты выдают центры сертификации, на подобие паспортных столов. При установке HTTPS-соединения первым делом проверяется подлинность сертификаты, и только потом происходит обмен данными.

Методы HTTP

Для начала необходимо понять из чего состоит запрос HTTP. Он состоит из URL (путь), который определяет сервер; метода, который определяет действие; куки, которые содержат данные сессий о клиенте.

Существует немало методов HTTP, но мы коснемся только 4 из них, а именно: GET, POST, PUT, DELETE. Забегая вперед, хочется сказать, что мы будем использовать только 3 метода из этих 4.

1. GET запрос. Означает, что мы хотим получить какой-нибудь определенный ресурс, например, HTML-файл, который содержит информацию о списке товаров.
2. POST запрос. Создать новые данные, например, добавить новые данные в базу данных.
3. PUT запрос. Обновить существующие данные.
4. DELETE запрос. Удалить определенные данные.

## База данных MongoDB

MongoDB – база данных, классический пример NoSQL-системы, использует схему базы данных. Написана на C++. Данные хранятся в документах в виде JSON. Чаще всего MongoDB хранит данные, которые не сильно связаны между собой. Получение информации происходит через запросы без SQL. Данные записываются в виде пар из ключей и значений. Благодаря такому формату в MongoDB можно разместить очень разнообразную и сложно структурированную информацию. MongoDB доступна всем, имеет открытый исходный код, и она бесплатная. СУБД подходит для операционных систем семейства Linux, Windows и macOS.

Как устроено хранение данных?

* Идентификатор. Он помогает обращаться к данным, именно по нему записи отличают друг от друга. Он должен быть в каждом документе, впрочем, он создается автоматически.
* Поле. Это одна запись в документе. Можно сравнить со столбцом в таблице реляционной базе данных.
* Документ — это файл в формате BSON: название расшифровывается как binary JSON. Отличие от стандартного JSON в том, что формат работает еще быстрее, но файлы в нем занимают меньше места.

Вместо традиционных понятий таблиц и полей в MongoDB используются свои сущности. Они отличаются от принятых в реляционных базах данных, но во многом схожи.

Так в чем же преимущества базы данных MongoDB?

1. Быстрота. Благодаря структуре документов, индексации и другим особенностям она работает быстро.
2. Гибкость. Позволяет хранить разнообразные данные, поэтому она и является гибкой.
3. Отсутствие сложных соединений. Конструкция join не используется. Нет необходимости описывать схему таблиц.
4. Легкая масштабируемость. Один из главных плюсов MongoDB, если информация должна измениться в базе данных, то не нужно полностью менять структуру всей базы данных.

Именно поэтому мы будем использовать эту базу данных.