**Проектирование программного средства**

**1.1 Библиотека React**

React — JavaScript-библиотека для работы с пользовательскими интерфейсами (UI), которую создали разработчики Facebook. Библиотеку начали использовать на сайте этой социальной сети в 2011 году. А в 2013 году Facebook открыл исходный код React.

С помощью React разработчики создают веб-приложения, которые изменяют отображение без перезагрузки страницы. Благодаря этому приложения быстро реагируют на действия пользователя, например, заполнение форм, применение фильтров, добавление товаров в корзину и так далее.

React применяют для отрисовки компонентов пользовательского интерфейса. Также библиотека может полностью управлять фронтендом. В этом случае React используют с библиотеками для управления состоянием и роутинга, например, Redux и React Router.

**1.2. Vite**

Для обеспечения плавной и быстрой разработки с минимальными предварительными настройками, Vite использует самые продвинутые функции современного JavaScript, такие как модули ES. Такие принципы позволяют делать как компиляцию фреймворка, так и использовать уже готовые пресеты. **Для установки не требуется наличия каких-то дополнительных плагинов, однако, в зависимости от специфики проекта.** Обычно установка производится совместно с одним-двумя плагинами, плюс, некоторая настройка под текущий проект.

**Основным преимуществом инструмента является простота добавления в проект и развертки.** Если разработчик знаком на среднем и более высоком уровне с JavaScript, а также Vue и React, то освоить работу с Vite.js для него не составит труда. К сожалению, документация хоть и обширная, но почти вся она на английском языке, поэтому полноценное освоение инструмента без знаний языка будет затруднительно. В русскоязычной среде разработчиков пока мало тех, кто активно пользуется Vite и при этом готов делиться опытом с начинающими разработчиками.

Если у вас есть база из навыков работы с [JS](https://liquidhub.ru/blogs/blog/js-razrabotchik), [React](https://liquidhub.ru/blogs/blog/chto-takoe-react) и [Vue](https://liquidhub.ru/blogs/blog/vue-js-dlya-veb-razrabotki), то освоить Vite вполне реально самостоятельно. Среда разработки, предоставляемая данным инструментом, не нуждается в обязательной настройке связки, но ее можно провести, если такая необходимость возникнет. **Это возможно благодаря обслуживанию за счет ES-модулей**, встроенных во все современные версии распространенных браузеров. Однако, если перед разработчиком стоит задача сделать сайт или приложение с поддержкой более старых версий браузеров, то Vite здесь может не подойти.

**1.3. NoSQL БД MongoDB**

MongoDB — это ориентированная на документы база данных NoSQL с открытым исходным кодом, которая использует для хранения структуру JSON. Модель данных MongoDB позволяет представлять иерархические отношения, проще хранить массивы и другие более сложные структуры.

Вместо таблиц и строк, как в реляционных базах данных, в MongoDB коллекции и документы, которые состоят из пар «ключ — значение»:

```

{

"someKey": "value",

"best": {

"foo": "baz"

}

```

Коллекция — это набор документов, эквивалент таблицы реляционной базы данных. Каждый документ может отличаться друг от друга размером, содержанием и количеством полей. В документах MongoDB можно хранить даже бинарные данные: изображения, mp3 и т. д.

Структура документа похожа на то, как разработчики конструируют классы и объекты на языках программирования, а у хранимых документов необязательно должна быть заранее определённая схема. Можно создавать поля на лету, а также настроить валидацию JSON Schema. Драйверы MongoDB уже из коробки умеют десериализовывать данные из базы в полноценные объекты. Для разработчика это делается прозрачно, а значит, требуется писать меньше кода и возникает меньше багов.

## 1.4. Язык разметки HTML5

Hyper Text Markup Language (HTML) – язык разметки гипертекста – предназначен для написания гипертекстовых документов, публикуемых в World Wide Web и позволяющих их там просматривать.

Гипертекстовый документ – это текстовый файл, имеющий специальные метки, называемые тегами, которые впоследствии опознаются браузером и используются им для отображения содержимого файла па экране компьютера.

Основным преимуществом гипертекста перед обычным текстом является возможность добавления к содержимому документа гиперссылок – специальных конструкций языка HTML, которые позволяют щелчком мыши перейти к просмотру другого документа.

HTML-документ состоит из двух частей: собственно, текста, т.е. данных, составляющих содержимое документа, и тегов – специальных конструкций языка HTML, используемых для разметки документа и управляющих его отображением. Теги языка HTML определяют, в каком виде будет представлен текст, какие его компоненты будут исполнять роль гипертекстовых ссылок, какие графические или мультимедийные объекты должны быть включены в документ.

## Язык CSS

CSS это язык стилей, определяющий отображение HTML-документов. Например, CSS работает с шрифтами, цветом, полями, строками, высотой, шириной, фоновыми изображениями, позиционированием элементов и многими другими вещами.

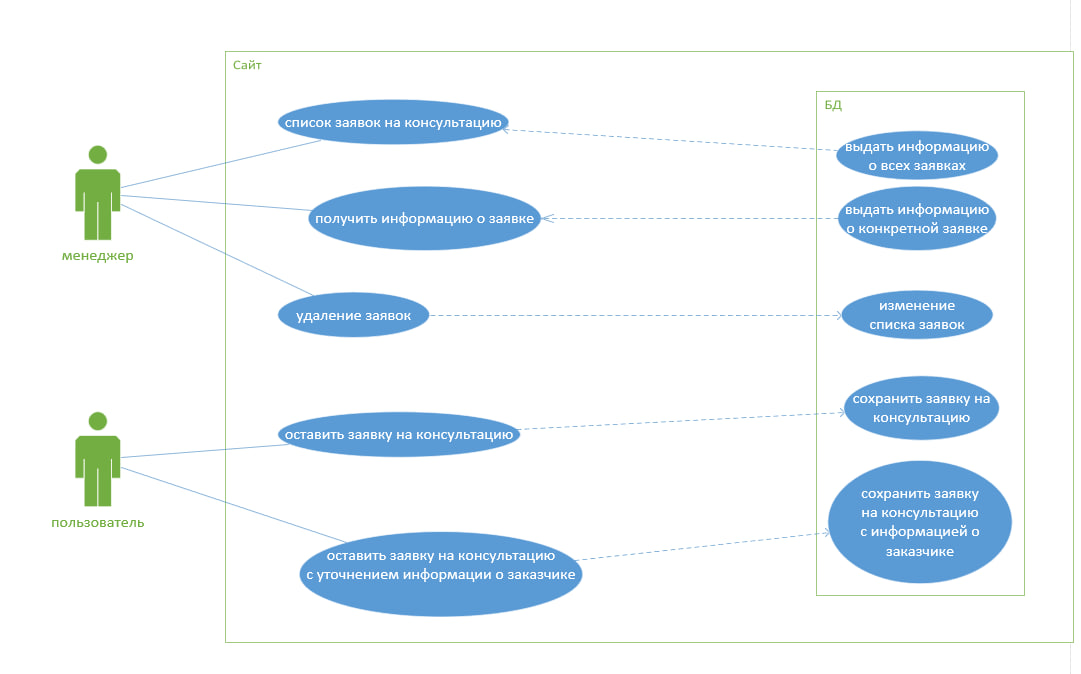
HTML используется для структурирования содержимого страницы. CSS используется для форматирования того же структурированного содержимого.

По мере развития Web дизайнеры начали искать возможности форматирования онлайновых документов. Чтобы удовлетворить возросшим требованиям потребителей, производители браузеров изобрели новые HTML-тэги, такие, например, как <font>, которые отличались от оригинальных HTML-тэгов тем, что они определяли внешний вид, а не структуру.

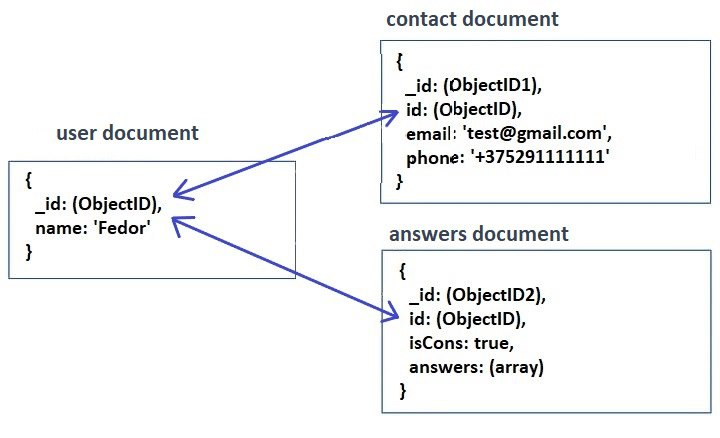
Это также привело к тому, что оригинальные тэги структурирования, такие как <table>, стали всё больше применяться для дизайна страниц вместо структурирования текста. Многие новые тэги дизайна, такие как <blink>, поддерживались только одним браузером.

CSS был создан для исправления этой ситуации путём предоставления web-дизайнерам возможностей точного дизайна, поддерживаемых всеми браузерами. Одновременно произошло разделение представления и содержимого документа, что значительно упростило работу.

## Диаграмма вариантов использования



## Структура БД



**Разработка программного средства**

Для разработки проекта были использованы следующие технические средства:

MongoDB - документоориентированная система управления базами данных типа NoSQL, не требующая описания схемы таблиц.

Webstorm IDE – редактор исходного кода, разработанный jetbrains.

Visual Studio Code - редактор исходного кода, разработанный Microsoft.

JavaScript - мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации ECMAScript. JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений.

Node.js - программная платформа, основанная на движке V8, превращающая JavaScript из узкоспециализированного языка в язык общего назначения.

React.js - JavaScript-библиотека с открытым исходным кодом для разработки пользовательских интерфейсов. React разрабатывается и поддерживается Facebook, Instagram и сообществом отдельных разработчиков и корпораций. React может использоваться для разработки одностраничных и мобильных приложений.

**Vite.js - это один из инструментов для настройки сред разработки.** Чаще всего используется в связке с фреймворками React и Vue.

Express.js - фреймворк web-приложений для Node.js, реализованный как свободное и открытое программное обеспечение под лицензией MIT. Он спроектирован для создания веб-приложений и API. Де-факто является стандартным каркасом для Node.js.

Axios.js - HTTP-клиент, основанный на [Promise](https://javascript.info/promise-basics) для [node.js](https://nodejs.org) и браузера.

Перехватывает запросы и ответы.

Cors.js - стандарт, позволяющий предоставлять веб-страницам доступ к объектам сторонних интернет-ресурсов.

Mongoose - библиотека объектно-ориентированного программирования JavaScript, которая создает соединение между MongoDB и средой выполнения JavaScript Node.js.

  React Router DOM — это модуль узла, который предназначен для маршрутизации в веб-приложениях.

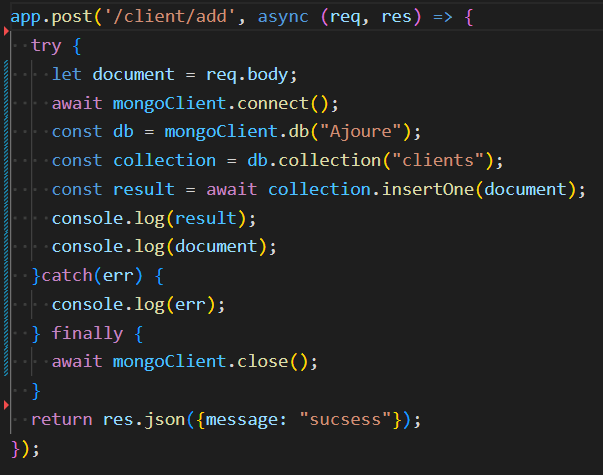
# Разработка серверной части

Серверная часть сайта была разработана в среде разработки Visual Studio Code. Для написания были использованы язык программирования JavaScript и модуль для NodeJS MongoDB.

Файл серверной части:



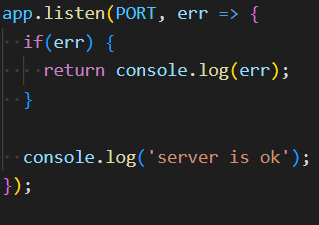
В файле back.js написан код всей серверной части сайта, а именно:

Описан POST-запрос для отправки документа в БД: ;

Описан GET-запрос для получения документов из БД:

;

Описана работы сервера Express.js:

.

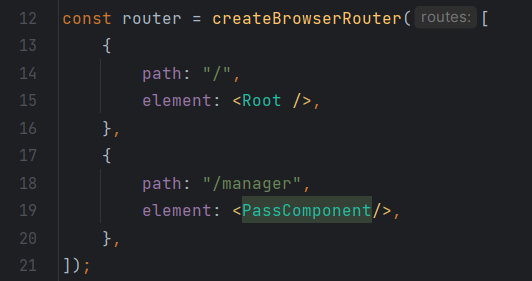
При отправке POST-запроса идёт получение документа для сохранения. Он берётся из тела POST-запроса. После идёт подключение к MongoDB, далее идёт выбор нужной коллекции. В итоге документ сохраняется в выбранную коллекцию в Базе Данных. Далее подключение с БД закрывается.

GET-запрос используется для админ-панели. После отправки GET-запроса со стороны администратора идёт подключение к БД, после чего загружаются все документы в переменную users, находящиеся там. Далее переменная с документами отправляется на сторону администратора и рендерится на странице, в результате чего администратор может видеть всех клиентов.

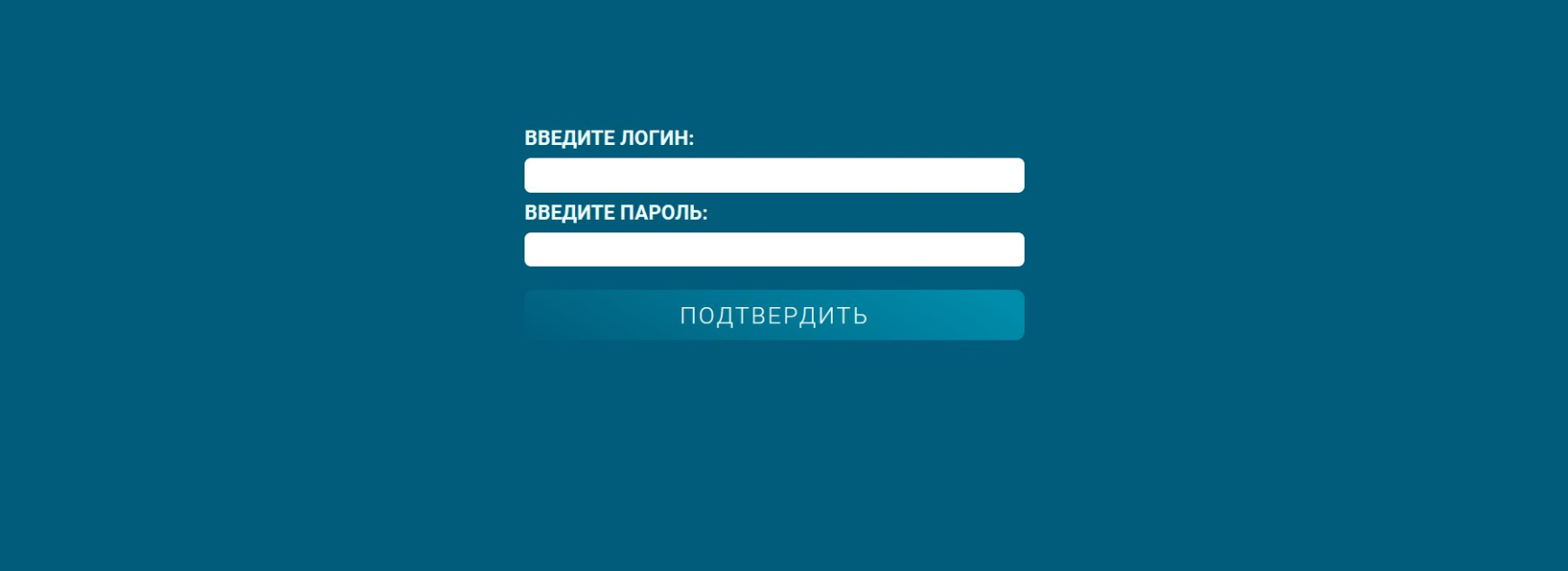
В описании функции работы сервера Express.JS используется порт 7777, на котором запускается сам сервер. Именно на этот порт и идёт отправка POST- и GET-запросов.

# Разработка части frontend

Разработка клиентской и менеджерской части приложения полностью велась в Webstorm IDE на ReactJS. Также использовался модуль react-router-dom. В данном веб-приложении есть 2 страницы пользовательская и админка.







Для того чтобы зайти в админку достаточно написать после названия сайта “/manager” и ввести логин (“admin”) и пароль (“admin”) после чего будет показана таблица входящих заказов. При нажатии на кнопку “информация” для каждого из клиентов, выбравших какие-то варианты ответов при отправке заявки будет отображаться информация, выбранная пользователем.

