Использовано:

На серверной стороне:

* Express – Фреймворк для работы с HTTP, для создания апи
* База данных – PostgreSQL
* Sequelize – ORM для работы с бд
* Cors – Для того, чтобы можно было с браузера обращаться к серверу
* Dotenv – для переменных окружения
* Nodemon – Перезапускает автоматически сервер при изменениях

На клиентской части:

* React – Фреймворк для созданию веб приложения
* Node-sass – пакет для работы с препроцессором sass
* Redux – Менеджер глобального хранилища
* React-router-dom – для работы с переходами по страницам
* Axios – для запросов на сервер

Структура проекта

Главная папка/ - Хранится клиентская и серверная часть

Главная папка/client – Клиентская часть приложения

/client/src – Папка, где находится все разработка клиентской части приложения

Главная папка/server – Серверная часть приложения

/server/index.js – Центральный файл для всех настроек и запуска сервера

/server/db.js – Файл для создания бд

/server/.env – файл для глобальных переменных

/server/models – Папка для создания таблиц бд

/server/routes – Папка с файлами для работы роутов(маршрутизация сервера)

/server/controllers – Папка с классами по работе веб приложения по определённым роутам(путям)

**Роуты к серверу(пути для обращения к серверу, маршрутизация сервера)**

**/statusorder**

Метод – get , путь - /statusorder. Получение статусов заказа

**/order/acrchive**

Метод – get, путь /order/archive. – Получение архивных заказов

**/orderinthehall**

Метод – get, путь /orderinthehall. – Получение заказов в зале

Метод – post, путь /orderinthehall. – Создание заказов в зал

Метод – delete, путь /orderinthehall/:id. – Удаление заказа в зале

**/takeawayorder**

Метод – get, путь /takeawayorder. – Получение заказов в с собой

Метод – post, путь /takeawayorder. – Создание заказов с собой

Метод – delete, путь /takeawayorder/:id. – Удаление заказа с собой

**/dish/type**

Метод – get, путь /dish/type. – Получение типов блюд

Метод – post, путь /dish/type. – Создание типов блюд

**/dish**

Метод – get, путь /dish. – Получение блюд

Метод – post, путь /dish. – Создание блюд

Метод – delete, путь /dish/:id. – Удаление блюд

Метод – put, путь /dish. – Изменение блюд

**/order**

Метод – get, путь /order?{Не обязательный параметр на сортировку type=date / type=status}. – Получение заказов

Метод – post, путь /order. – Создание заказов

Метод – put, путь /order. – Изменение блюд

Метод – delete, путь /order/dish/:id. – Удаление блюда из заказа

Метод – post, путь /order/dish. – Создание блюда к заказу

**Классы**

class DishController класс по работе с блюдами

Его методы:

* create – Создание блюда
* getAll – Получение блюд
* update – Изменение блюда
* delete – Удаление блюда

class DishTypesController класс по работе с типа блюд

Его методы:

* create – Создание типа блюда
* get – Получение типов блюд

class OrderArchive класс по работе с архивными заказами

Его методы:

* getAll – Получение архивных заказов

class OrderController класс по работе с заказами

Его методы:

* create – Создание заказа
* getAll – Получение заказов
* update – Изменение заказа
* deleteDish – Удаление блюда из заказа
* addDishToOrder – Добавление блюда в заказ

class OrderInTheHallController класс по работе с заказами в зал

Его методы:

* create – Создание заказа в зал
* update – Изменение заказа в зал
* delete – Удаление заказа в зал

class OrderStatusController класс по работе с статусами заказов

Его методы:

* create – Создание статуса заказа
* getAll – Получение статусов заказов

class TakeAwayOrderController класс по работе с заказами с собой

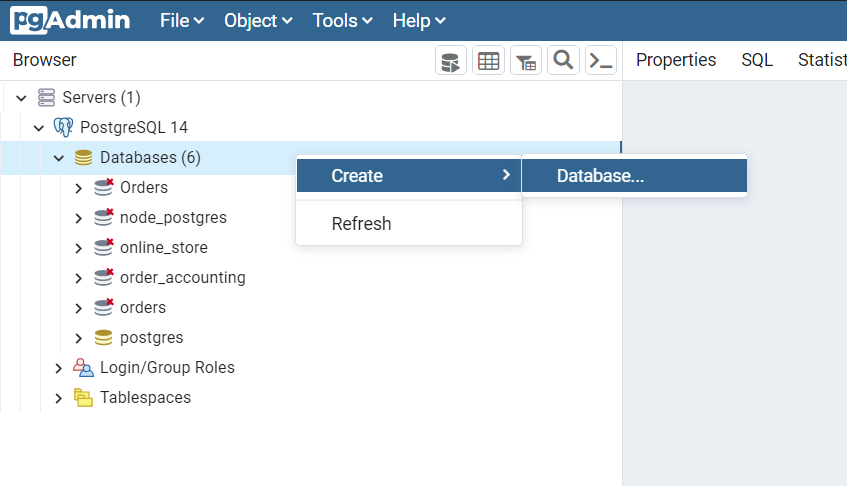
Его методы:

* create – Создание заказа с собой
* update – Изменение заказа с собой
* delete – Удаление заказа с собой

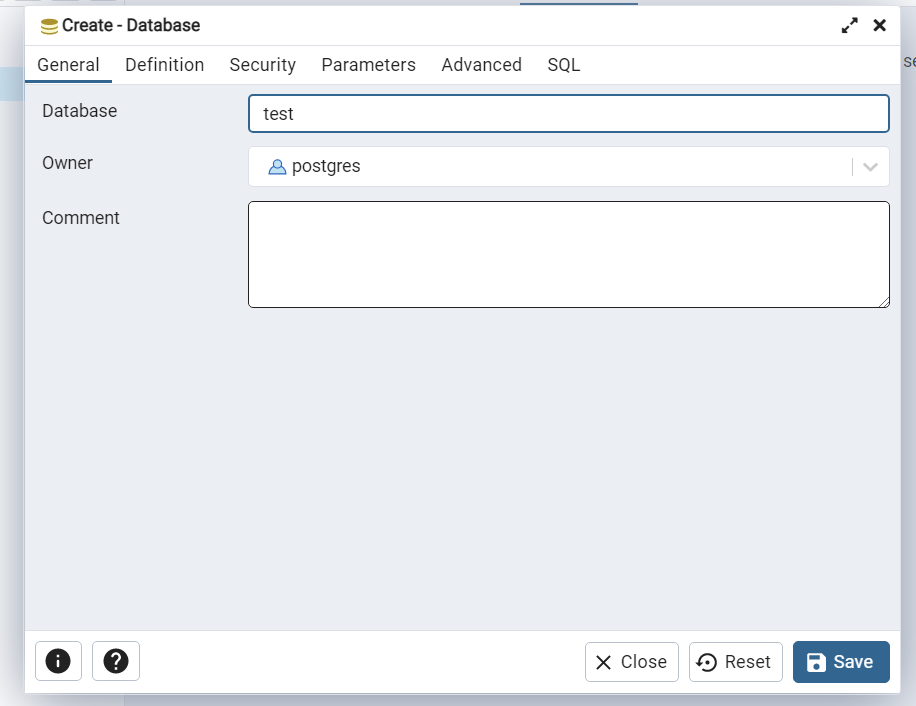
**Запуск проекта**

Для начала нужно скачать pgAdmin, при установке данного ПО надо запомнить данные user(по умолчанию - postgres), port(по умолчанию - 5432), **password**.

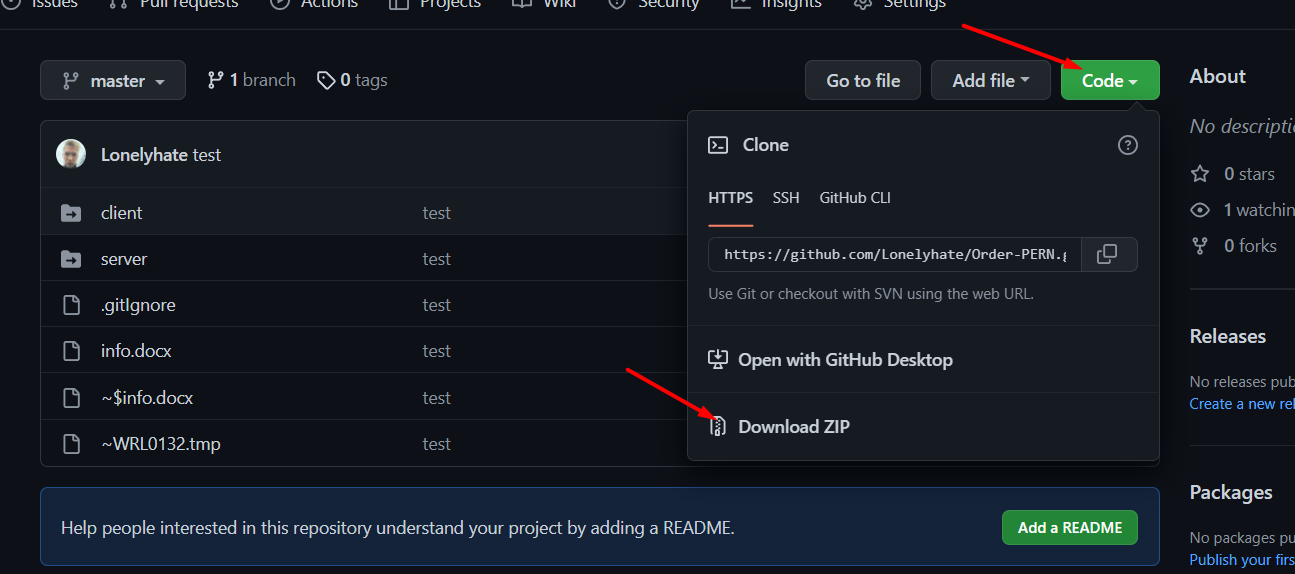
После установки нужно создать бд для сервера



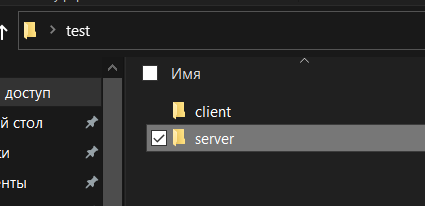
Создаем бд



В репозитории скачиваем архив

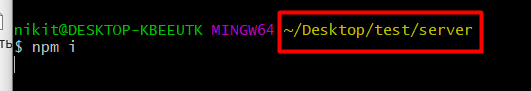


Создаем папку для проекта и переносим туда папку сервер и клиент



Заходим в папку сервер

Открываем консоль и пишем команду npm i



После успешной установки зависимостей точно также проделываем и в папке client



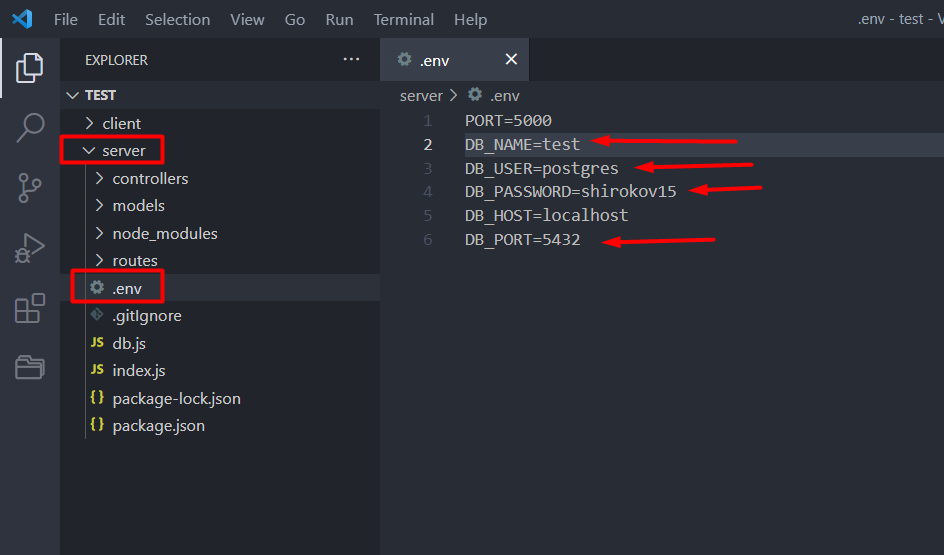
Далее открываем данный проект с помощью редактора какого нибудь и переходим в файл .env для подключение бд

DB\_NAME = [Вписать имя бд, которое вы создали]

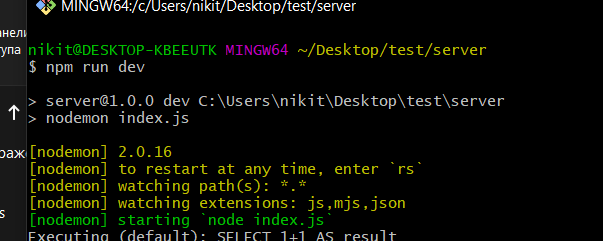
DB\_USER = [По идеи можно не исправлять]

DB\_PASSWORD = [Пароль при установке ПО pgAdmin]

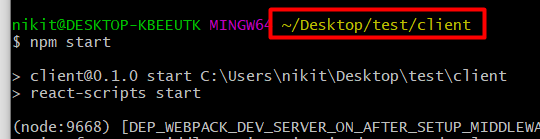
DB\_PORT = [По умолчанию должен был быть 5432, поэтому менять не надо]



Далее открываем консоль в папке сервер и прописываем npm run server. Не закрываем данную консоль, пока находитесь на сайте, благодаря ней работает сервер



Теперь запускаем клиент, точно также заходим в папку client и открываем в ней консоль и прописываем в ней npm start, точно также не закрываем консоль, потому что она работу осуществляет клиента. Данная команда автоматически открывает страницу в браузере с приложением



Все, приложение запущено

