**Сайт «Онлайн доска»**

**Руководство пользователя**

**Содержание**

[1 Введение 2](#_Toc181120925)

[1.1 Наименование системы 2](#_Toc181120926)

[1.2 Область применения 2](#_Toc181120927)

[1.3 Условные обозначения системы 2](#_Toc181120928)

[2 Установка системы 3](#_Toc181120929)

[2.1 Требование для установки 3](#_Toc181120930)

[2.2 Установка зависимостей 3](#_Toc181120931)

[2.3 Настройка базы данных 3](#_Toc181120932)

[2.4 Запуск сервера 3](#_Toc181120933)

[3 Внутреняя структура системы 4](#_Toc181120934)

[3.1 Используемый технологический стек 4](#_Toc181120935)

[3.2 Функции и их описание 4](#_Toc181120936)

[3.3 Структура данных 4](#_Toc181120937)

[3.4 Описание API 6](#_Toc181120938)

# Введение

## Наименование системы

Сайт «Онлайн доска».

## Область применения

«Онлайн доска» представляет собой веб-приложение, предназначенное для организации дистанционного обучения и совместной работы в режиме реального времени. Основная цель «Онлайн доски» - обеспечить удобную и эффективную платформу для проведения онлайн-занятий, презентаций и совместной работы над учебными материалами.

## Условные обозначения системы

Условное обозначение сайта «Онлайн доска» - Сайт.

# Установка системы

## Требование для установки

Для развертывания Сайта нужен сервер, который поддерживает Node.js и так же PostgreSQL. И так же на сервер должны установлены следующие библиотеки: Canvas API и т.д.

Так же Сайт должен поддерживать все современные браузеры, в числе которых есть: Chrome, Firefox, Safari и т.д.

## Установка зависимостей

Фронтенд: установите зависимости для фронтенда (если есть), выполнив следующую команду:

npm install

Бэкенд: установите необходимые библиотеки node.js (если применимо), например, через Composer:

npm install expres

## Настройка базы данных

Чтоб установить базу данных на сервер нужно создать новую базу данных в PostgreSQL и импортировать готовую структуру из уже существующей базы данных.

## Запуск сервера

Чтоб запустить сервер, в папке проекта нужно открыть терминал и вписать следующую команду:

npm start

# Внутреняя структура системы

## Используемый технологический стек

Сайт использует для фронтенда HTML и CSS, для бекэнда используется Node.js с библиотекой Canvas API. В качестве базы данных используется PostgreSQL

## Функции и их описание

У Сайта имеются следующие функции:

* создать доску – создает уникальный URL с доской который в последующем передается пользователю, так же в базе данных создается новая запись с названием доски и URL ссылкой на неё;
* пригласить пользователей на созданную доску – создает пригласительную ссылку, с помощью которой другие пользователи смогут присоединиться к уже созданной доске;
* дать определённому пользователю доступ к редактированию доски – в базе данных редактируется запись что в дальнейшем дает выбранному пользователю редактировать доску;
* выгнать определённого пользователя с доски – в базе данных удаляется запись с выбранным пользователем, после чего этот пользователь теряет доступ к просмотру и редактированию доски;
* выбрать толщину «кисти» – это функция работает с помощью библиотеки Canvas API;
* выбрать цвет «кисти» – это функция работает с помощью библиотеки Canvas API;
* нарисовать на доске – это функция работает с помощью библиотеки Canvas API;
* выбрать толщину ластика – это функция работает с помощью библиотеки Canvas API;
* стереть с доски – это функция работает с помощью библиотеки Canvas API;
* добавить текст на доску – это функция работает с помощью библиотеки Canvas API;
* добавить изображение формата «.png» и «.jpg» на доску – это функция работает с помощью библиотеки Canvas API.

## Структура данных

Сайт имеет база данных, структура этой базы данных изображена в виде логической ER на рисунке 1 и физической ER которая изображена на рисунке 2 соответственно.

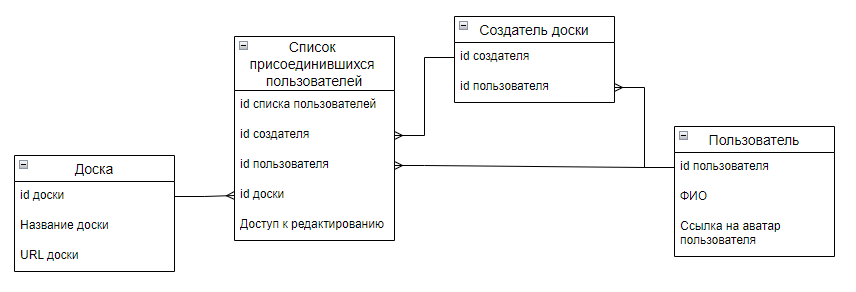


Рисунок 1 – Логическая ER диаграмма

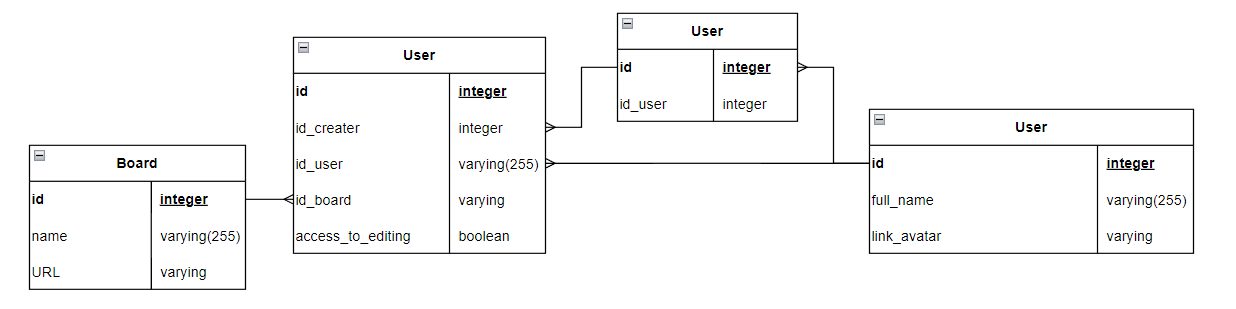


Рисунок 2 – Физическая ER диаграмма

На ER диаграмма показано что есть 3 таблицы: «Доски», «Создатель доски» и «Пользователи».

Таблица «Доски» содержит следующие данные:

* id доски: Идентификатор доски.
* Название доски: название доски
* Ссылка на доску: URL-адрес, по которому можно найти доску.

Таблица «Создатель доски»

* id создателя доски: Уникальный идентификатор создателя доски.
* id пользователя: Ссылка на пользователя, который создал доску (внешний ключ к таблице «Пользователи»).

Таблица «Пользователи»

* id пользователя: Уникальный идентификатор пользователя.
* ФИО: Полное имя пользователя.
* Ссылка на аватар пользователя: URL-адрес изображения аватара пользователя.

Таблица «Список присоединившихся пользователей»

* id списка пользователей: Уникальный идентификатор для списка пользователей, которые присоединились к конкретной доске.
* id создателя: Ссылка на таблицу «Создатель доски», которая идентифицирует присоединившегося создателя доски.
* id пользователя: Ссылка на таблицу «Пользователи», которая идентифицирует присоединившегося пользователя.
* id доски: Ссылка на таблицу «Доски», указывающая на доску, к которой пользователь присоединился.
* Доступ к редактированию: разрешение, которое указывает, имеет ли пользователь права на редактирование доски.

## Описание API

На Сайте используются следующие API запросы:

* POST - запрос – используется для функции «создании доски»;
* GET - запрос – используется для получение данных с доски;
* PATCH - запрос – используется для редактирование прав доступа пользователей а так же для функции «выгнать пользователя с доски»