| Продольный бланк_ВятГУ_распорядительный акт+ | | | |
| --- | --- | --- | --- |
|  | |  | |
|  |  |  |  |
|  |  |  |  |
|  | | | |
| **Руководство программиста**  веб-приложения “NeuroKanban”  2023 | | | |
|  | | | |

**Содержание**

[**1. Общие сведения о программе 3**](#_gjdgxs)

[1.1 Назначение программы 3](#_v6g3cfm7n9us)

[1.2 Функциональное назначение программы 3](#_1x4fmo11szxr)

[**2. Установка программы и развертывание 3**](#_xfxtfdpcdvmo)

[2.1. Минимальные требования к программным средствам 3](#_habl6rlic109)

[2.2. Минимальные требования к программным средствам 4](#_wdkmpa1tugrm)

[2.3. Требования к персоналу 4](#_dhv41f90isd1)

[2.4. Процесс развертывания 4](#_1krhw5o3mp84)

[2.5. Процесс разработки 5](#_d2yywuki1ocg)

[**3. Структура системы 7**](#_3jnvxawwbwsf)

[3.1. Составные части системы 7](#_busy47got1nw)

[3.2. Файловая структура системы 7](#_6ocughj5k9lw)

# **Общие сведения о программе**

## **Назначение программы**

Система “NeuroKanban” предназначена для организации работы над проектами, необходимо обеспечивать распределение задач по участникам и стадиям выполнения. Для упрощения этого процесса и реализуется данный проект, а именно: обеспечение платформы для управления проектами небольших проектных групп, используя парадигму канбан.

## **Функциональное назначение программы**

* создание и редактирование проектов в системе;
* создание и редактирование «досок» внутри проектов;
* создание и редактирование групп задач (столбцов) на «досках»;
* создание, редактирование, перемещение задач; между группами на «досках», а также установление различных характеристик задач, таких как трудоемкость (предполагаемые время исполнения, крайняя дата выполнения, уже затраченное время исполнения);
* обеспечение напоминаний о заканчивающемся сроке для выполнения задач;
* информирование руководителя проекта о статусе выполнения задач.

# **Установка программы и развертывание**

## **Минимальные требования к программным средствам**

Для полноценного функционирования системы “NeuroKanban” необходимы:

* система сборки и развертывания контейнерных приложений Docker;
* инструмент docker compose для Docker;
* программа запуска JavaScript приложений Node.JS версии 18.18.2;
* пакетный менеджер pnpm для Node.JS.

## **Минимальные требования к программным средствам**

Для полноценного развертывания системы “NeuroKanban” необходим сервер, имеющий следующие технические характеристики:

* операционная система GNU/Linux Ubuntu 22.04;
* процессор Intel Pentium G4560 (@3.5 2/4);
* ОЗУ 4gb DDR4 2400mhz;
* клавиатура;
* место на накопителе – 1024 мб;
* наличие интернет-соединения со стабильной скоростью передачи данных не ниже 1 Мбит/с.

## **Требования к персоналу**

Программист должен иметь минимум среднее техническое образование.

В перечень задач, выполняемых программистом, должны входить:

* задача установки и поддержания работоспособности программных средств;
* задача диагностики и определения причин неисправности web-сайта.

## **Процесс развертывания**

Для удобства работы с docker был создан Makefile, при помощи команд которого можно выполнить различные действия с системой.

Развертывание включает в себя:

1. запуск на сервере команды make migrate для создания файла БД с необходимой структурой;
2. запуск на сервере команды make build для сборки контейнерного решения;
3. запуск на сервере команды make run для запуска системы.

Таким образом в результате запуска будет развернута система на 80 (клиентская часть) и 3500 (серверная часть) портах сервера, где производился запуск. При обновлении кода необходимо выполнить заново 2 и 3 пункт, предварительно остановив сервер командой make down.

## **Процесс разработки**

Для разработки необходимо выполнить команды запуска системы в режиме разработки. Процесс включает в себя:

1. запуск команды make migrate для создания файла БД с необходимой структурой (выполняется единоразово);
2. запуск команды pnpm install –frozen-lockfile для работы в средах разработки с подсказками кода;
3. запуск команды make build\_dev для сборки контейнерного решения;
4. запуск команды make run\_dev для запуска системы.

Таким образом в результате запуска будет развернута система на 3000 (клиентская часть) и 3500 (серверная часть) портах компьютера, где производился запуск системы в режиме разработки. Docker настроен так, что при изменении файлов и их сохранении, нет необходимости в сборке контейнера заново и его перезапуска. Будет происходить так называемая динамическая подгрузка кода.

# **Структура системы**

## **Составные части системы**

Система разделяется на две части: клиентскую (фронтенд) и серверную (бэкенд).

Бэкенд представляет собой API, к которому происходит обращение от клиентской части. Также он работает с базой данных.

Фронтенд же представляет собой веб-сервер, реализующий пользовательский интерфейс системы для пользователей. Производит получение данных от пользователей, отправку на бэкенд и вывод ответа сервера.

## **Файловая структура системы**

Система представлена в виде двух папок frontend и backend, в которых реализована клиентская часть и серверная часть соответственно.

Серверная часть представлена в виде следующих типов данных:

* “ts” - файлы TypeScript, содержащие в себе логику серверной части, такую как маршруты, классы и т.д.;
* “prisma” - файлы Prisma для описания структуры базы данных.

Клиентская часть представлена в виде следующих типов данных:

* “tsx” - файлы описания компонентов React на языке TypeScript XML;
* “ts” - файлы TypeScript с дополнительной логикой, используются tsx файлами;
* “scss” и “css” - файлы описания стилей веб-страниц.