

УТВЕРЖДАЮ

Руководитель практики от Университета

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(должность, ФИО)*

МП \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(Подпись)*

Дата « \_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2023 год

**ОТЧЕТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

ПМ.03 Ревьюирование программных продуктов

|  |  |
| --- | --- |
| Елпашев Сергей Антонович | |
| *(Ф.И.О. обучающегося)* | |
| 09.02.07 Информационные системы и программирование | |
| *(специальность)* | |
|  | |
| Учебная группа | ИСПк-402-52-00 |
|  |  |
| Место прохождения практики | ФГБОУ ВО «Вятский государственный университет», |
| Колледж ВятГУ | *(наименование организации, структурного подразделения организации)* |
|  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Итоговая оценка: |  | | | | |
| Руководитель  практики от университета |  |  |  |  |  |
|  | *(дата)* |  | *(подпись)* |  | *(Ф.И.О.)* |

Киров, 2023 г.



ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ НА ПРАКТИКУ

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ф.И.О. обучающегося | | | Елпашев Сергей Антонович | | | | | | |
| Специальность | | | 09.02.07 Информационные системы и программирование | | | | | | |
| Учебная группа | | | ИСПк-402-52-00 | | | | | | |
| Вид практики | | | учебная практика | | | | | | |
| Сроки прохождения практики с | | | |  | по |  | | | |
| Место прохождения практики | | | | ФГБОУ ВО «Вятский государственный университет»,  Колледж ВятГУ | | | | | |
|  | | (наименование организации, структурного подразделения организации) | | | | | | |
| № | Виды работ, выполняемых обучающимися во время практики | | | | | | Объем работ (час) | Формируемые компетенции | | |
| 1 | Пройти инструктаж по ознакомлению с правилами внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологическими правилами и гигиеническими нормативами, а также вводный инструктаж и инструктаж на рабочем месте | | | | | | 2 | ОК-7 | | |
| 2 | Определение требований к программному продукту и его функциональных характеристик, поиск и анализ готовых технических решений | | | | | | 4 | ОК-1-4 | | |
| 3 | Разработка технической документации на программный продукта | | | | | | 10 | ОК-1, ОК-2, ОК-4, ПК-3.3 ОК-9-11, ПК-3.2, ПК-3.3 | | |
| 4 | Ревьюирование программного кода. Создание репозитория | | | | | | 4 | ОК-1, ОК-4, ОК-8, ОК-10, ПК-3.1, ПК-3.4 | | |
| 6 | Разработка эксплуатационной документации | | | | | | 4 | ОК-10, ПК-3.4, ПК-3.5 | | |
| 7 | Подготовка презентации для представления программного продукта потенциальному потребителю. | | | | | | 4 | ОК-5 | | |
| 8 | Подготовка к сдаче и сдача промежуточной аттестации | | | | | | 2 | ОК-6 | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Индивидуальное задание на практику разработано в соответствии с рабочей программой практики. | | | | | |
| Руководитель  практики от университета |  |  |  |  |  |
|  | (дата) |  | (подпись) |  | (Ф.И.О.) |

|  |  |
| --- | --- |
| С индивидуальным заданием ознакомлен(а) |  |
|  | (дата, подпись обучающегося) |

**ОТЗЫВ РУКОВОДИТЕЛЯ ПРАКТИКИ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ф.И.О. обучающегося | | Елпашев Сергей Антонович | | | |
| Специальность | | 09.02.07 Информационные системы и программирование | | | |
| Учебная группа | | ИСПк-402-52-00 | | | |
| Вид практики | | учебная практика | | | |
| Сроки прохождения практики с | | | 15.09.2023 | по | 20.11.2023 |
| Место прохождения практики | | | ФГБОУ ВО «Вятский государственный университет», Колледж ВятГУ | | |
|  | (наименование организации, структурного подразделения организации) | | | | |

ВИДЫ И КАЧЕСТВО ВЫПОЛНЕНИЯ РАБОТ

|  |  |  |  |
| --- | --- | --- | --- |
| Вид работ | Критерий выполнения работ | | |
| Выполнены полностью самостоятельно | Выполнены с незначительной помощью наставника | Выполнены с помощью наставника |
| Определение требований к программному продукту и его функциональных характеристик, поиск и анализ готовых технических решений | V |  |  |
| Разработка технической документации на программный продукта | V |  |  |
| Ревьюирование программного кода. Создание репозитория | V |  |  |
| Разработка эксплуатационной документации |  |  |  |
| Подготовка презентации для представления программного продукта потенциальному потребителю. | V |  |  |
| Подготовка к сдаче и сдача промежуточной аттестации | V |  |  |

Обучающийся ознакомлен с правилами внутреннего трудового распорядка, охраны труда, техники безопасности, противопожарной безопасности, санитарно-эпидемиологическими правилами и гигиеническими нормативами, а также прошел вводный инструктаж и инструктаж на рабочем месте.

Во время прохождения учебной практики обучающимся освоены следующие профессиональные и общие компетенции:

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование компетенции | Показатели оценки | Оценка | |
| Освоена | Не освоена |
| ПК 3.1. Осуществлять ревьюирование программного кода в соответствии с технической документацией. | Способен анализировать программный код с целью выявления некачественных архитектурных решений и критических мест в программе | V |  |
| ПК 3.2. Выполнять процесс измерения характеристик компонент программного продукта для определения соответствия заданным критериям | Способен верифицировать компоненты программного обеспечения в соответствии с заданными критериями | V |  |
| ПК 3.3. Производить исследование созданного программного кода с использованием специализированных программных средств с целью выявления ошибок и отклонения от алгоритма. | Способен готовить тесты для осуществления автоматизированного выявления ошибок в разрабатываемом программном обеспечении | V |  |
| ПК 3.4. Проводить сравнительный анализ программных продуктов и средств разработки, с целью выявления наилучшего решения согласно критериям, определённым техническим заданиям. | Способен подбирать средства разработки ПО наиболее подходящие по критериям определенным в техническом задании. | V |  |
| ПК 3.5. Проводить исследование проектной документации программного модуля. | Способен разрабатывать техническую и эксплуатационную документацию на программное обеспечение | V |  |
| ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам. | Способен оценивать предметную область и выбирать оптимальные способы решения задач профессиональной деятельности | V |  |
| ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности. | Способен анализировать специализированную информацию и находить оптимальные пути решения задач профессиональной деятельности | V |  |
| ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие. | Способен находить пути улучшения имеющихся решений, позволяющих повысить их общий качественный уровень | V |  |
| ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами. | Способен грамотно формулировать запросы в целях получения разъясняющей информации | V |  |
| ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста. | Способен разрабатывать проектную, техническую и пользовательскую документации | V |  |
| ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей. | Способен конструктивно обмениваться информацией с коллегами, демонстрировать осознанное поведение в ходе выполнения проектных работ | V |  |
| ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях. | Способен прогнозировать эффективность и ресурсозатратность используемых средств | V |  |
| ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности. | Способен соблюдать требования внутреннего трудового распорядка организации, охраны труда и техники безопасности в целях сохранения собственного здоровья | V |  |
| ОК 09. Использовать информационные технологии в профессиональной деятельности. | Способен применять современные инструменты создания ПО, в том числе для осуществления коллективной работы. | V |  |
| ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языке. | Способен использовать в своей работе специализированную документацию | V |  |
| ОК 11. Планировать предпринимательскую деятельность в профессиональной сфере. | Способен разрабатывать презентации для представления программного продукта потенциальному потребителю. | V |  |

**Краткая характеристика работы обучающегося**

|  |
| --- |
| Программа практики выполнена в полном объеме. Все виды работ выполнялись в срок, |
| без существенных замечаний. |
|  |

|  |  |  |
| --- | --- | --- |
|  |  | Руководитель практики  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Подпись ФИО  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (должность)  Дата «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2023 год |

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc150202999)

[**1. СВЕДЕНИЯ О РАБОТЕ, ВЫПОЛНЕННОЙ В ПЕРИОД ПРОХОЖДЕНИЯ УЧЕБНОЙ ПРАКТИКИ** 4](#_Toc150203000)

[**2. АНАЛИЗ ПОСТАВЛЕННОЙ ЗАДАЧИ** 5](#_Toc150203001)

[**3. РАЗРАБОТКА ТЕХНИЧЕСКОГО ЗАДАНИЯ** 7](#_Toc150203002)

[**4. Разработка Технического проекта** 8](#_Toc150203003)

[**5. ИСПОЛЬЗУЕМЫЙ ИНСТРУМЕНТАРИЙ И РАБОЧЕЕ ОКРУЖЕНИЕ** 9](#_Toc150203004)

[**ЗАКЛЮЧЕНИЕ** 10](#_Toc150203005)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 11](#_Toc150203006)

[**Приложение А** 12](#_Toc150203007)

[**ПРИЛОЖЕНИЕ Б** 25](#_Toc150203008)

[**ПРИЛОЖЕНИЕ В** 39](#_Toc150203009)

[**ПРИЛОЖЕНИЕ Г** 44](#_Toc150203010)

[**ПРИЛОЖЕНИЕ Д** 57](#_Toc150203011)

[**ПРИЛОЖЕНИЕ Е** 67](#_Toc150203012)

# **ВВЕДЕНИЕ**

Учебная практика ПМ.07 проходила на базе Колледжа ФГБОУ ВО «Вятский государственный университет» в период с 15.09.2023 по 20.11.2023.

Цель практики: сформировать у обучающихся навыки разработки программного обеспечения, как законченного продукта с размещением артефактов на онлайн-хостинге.

Задачи практики:

– закрепить полученные в ходе освоения предшествующих дисциплин навыки и умения в области создания программных продуктов;

– закрепить навыки анализа кода с целью выявления неэффективных решений;

– закрепить навыки разработки технической и эксплуатационной документации.

# **1. СВЕДЕНИЯ О РАБОТЕ, ВЫПОЛНЕННОЙ В ПЕРИОД ПРОХОЖДЕНИЯ УЧЕБНОЙ ПРАКТИКИ**

В период 15.09.2023 по 20.11.2023 при прохождении учебной практики ПМ.06 на базе ФГОБУ ВО «Вятский государственный университет» был выполнен следующий перечень работ, представленный в таблице 1.

Таблица 1 – Сведения о работе, выполненной в период практики

| Дата | Краткое содержание выполненных работ |
| --- | --- |
| 15.09.2023 | Ознакомление с правилами внутреннего трудового распорядка организации, требованиями охраны труда и техники безопасности |
| 15.09.2023-29.09.2023 | Подготовка аналитической записки с указанием цели, назначения и функциональных характеристик разрабатываемого программного продукта |
| 29.09.2023-13.10.2023 | Подготовка технического проекта содержащего описание структуры и алгоритмических решений применяемых в программном продукте |
| 13.10.2023-27.10.2023 | Разработка эксплуатационной документации |
| 27.10.2023-10.11.2023 | Подготовка презентации программного продукта и окончательное формирование репозитория. |
| 20.11.2023 | Подготовка к сдаче и сдача промежуточной аттестации |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата) (подпись)

# **2. АНАЛИЗ ПОСТАВЛЕННОЙ ЗАДАЧИ**

Наименование проекта: веб-программа NeuroKanban.

Целевая аудитория: небольшие проектные группы людей.

Назначение разработки: организация работы над проектами, обеспечение распределения 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.

Описание взаимодействия с потенциальным пользователем:

Потенциальный пользователь после попадания на приветственную страницу веб-сайта может перейти на страницу авторизации, после которой попадает на экран выбора проектов, имея также возможность создать проект в системе. После выбора/создания проекта пользователь может создать, либо взаимодействовать с досками внутри данного проекта. После выбора/создания «доски» пользователь может создать, либо взаимодействовать с задачами, размещенными в группах. При желании могут быть созданы новые группы задач, либо удалены/отредактированы уже имеющиеся. Задачи могут быть перемещены с одной группы задач в другую группу, либо же отредактированы/удалены с доски. Если же пользователь был приглашен в проект и не был назначен руководителем проекта, то у него есть возможность взаимодействовать с задачами внутри «досок», но не создавать новые «доски»/задачи.

Возможные аналоги:

* Trello - облачная программа для управления проектами небольших групп, разработанная Fog Creek Software. Минусы:

Прекращение работы на территории Российской Федерации;

Наличие платного функционала;

Коммерческий проект с закрытым исходным кодом;

* YouTrack — коммерческая система отслеживания ошибок, программное обеспечение для управления проектами, разработанное компанией JetBrains. Минусы:

Коммерческий проект с закрытым исходным кодом;

Наличие платного использования для более 10 человек, либо закрытого проекта;

Отсутствие возможности приобретения платного функционала на территории Российской Федерации;

* Jira — коммерческая система отслеживания ошибок, предназначена для организации взаимодействия с пользователями, хотя в некоторых случаях используется и для управления проектами. Минусы:

Проприетарное программное обеспечение с закрытым исходным кодом;

Прекращение работы на территории Российской Федерации;

Наличие платного функционала;

* Kaiten - российский сервис для совместной работы команд. Минусы:

Проприетарное программное обеспечение с закрытым исходным кодом;

Наличие платного функционала.

# **3. РАЗРАБОТКА ТЕХНИЧЕСКОГО ЗАДАНИЯ**

В ходе данного этапа было разработано техническое задание на основе функциональных требований к конечному продукту, включающие в себя следующее:

* навигация на сайте посредством пользовательского интерфейса;
* создание и редактирование проектов в системе;
* приглашение пользователей в проект от руководителей проектов;
* 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 (предполагаемое время исполнения, крайняя дата выполнения, уже затраченное время исполнения);
* обеспечение напоминаний о заканчивающемся сроке для выполнения задач;
* информирование руководителей проекта о статусе выполнения задач.

Технологическим стеком системы является: TypeScript React + SCSS на пользовательской части и Express + Prisma + socket.io на серверной части, развернутой на веб-сервисе Nginx и СУБД SQLite. Вся система разворачивается при помощи Docker контейнеров.

Подробное описание представлено в приложении А (текст технического задания).

# **4. Разработка Технического проекта**

В ходе данного этапа было разработан технический проект, где были описаны:

* Структура входных и выходных данных;
* Форма представления входных и выходных данных;
* Архитектура системы;
* Алгоритмы решения задач;
* Определение языка, структуры программы и требований к техническим средствам.

Входными данными являются:

* ввод данных авторизации;
* ввод данных проекта
* ввод данных доски проекта;
* ввод данных группы задач доски;
* ввод данных задачи;
* ввод почты пользователя для приглашения;

Выходными данными являются:

* вывод проектов после авторизации;
* вывод досок проекта;
* вывод групп задач доски;
* вывод задач;
* вывод участников проекта.

Архитектурой в данном программном продукте является двухзвенная архитектура “клиент-сервер”. Подразумевается взаимодействие множества клиентов с сервером посредством “тонкого” клиента - веб-сайта у множества клиентов, подключаемых к одному серверу обработки данных с СУБД и БД.

Подробное описание представлено в приложении Б (текст технического проекта).

# **5. ИСПОЛЬЗУЕМЫЙ ИНСТРУМЕНТАРИЙ И РАБОЧЕЕ ОКРУЖЕНИЕ**

В ходе разработки данного программного продукта был использован следующий инструментарий:

* язык программирования TypeScript;
* язык разметки html;
* языки декорирования: SCSS, CSS;
* библиотеки и фреймворки: React, Prisma, Socket.io;
* плагин форматирования кода Prettier;
* программное обеспечение для развертывания контейнерных приложений Docker;
* СУБД SQLite.

В качестве среды разработки использовался текстовый редактор Visual Studio Code.

В качестве системы контроля версий в соответствии с заданием на практику используется Git. Сам репозиторий располагается на в системе Github по ссылке <https://github.com/MrSedan/neuro-trello>.

Состав репозитория:

* папка .vscode – папка настроек редактора кода для проекта;
* папка frontend – папка с исходным кодом клиентской части;
* папка backend – папка с исходным кодом серверной части;
* файл .prettierrc – файл конфига для плагина форматирования кода Prettier;
* файл Makefile – файл с инструкциями сборки системы;
* README.md – файл с кратким описанием проекта и инструкциями развертывания;
* Файлы docker-compose – файлы описания инструкций сборки проектов с несколькими контейнерами Docker.

# **ЗАКЛЮЧЕНИЕ**

В ходе прохождения учебной практики ПМ.03 была ревьювирована веб-программа NeuroKanban для управления проектами небольших проектных групп, используя парадигму канбан.

Были улучшены и закреплены навыки разработки и работы с документацией программных продуктов. Также были освоены навыки модификации существующих программных модулей и создание дополнительных модулей для работы уже существующего. Был освоен фреймворк React для реализации клиентской части для отображения данных. Закреплены знания в области разработки на языке TypeScript, верстки и дизайна веб-сайтов и работы с СУБД SQLite.

В ходе ревьюирования были разработаны следующие документы: аналитическая записка, техническое задание, технический проект, руководство программиста, руководство пользователя, программа и методика испытаний. В ходе разработки документов были закреплены навыки разработки документации по программному продукту и навыки использования ГОСТа.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. TypeScript Documentation. — Текст : электронный // TypeScript : [сайт]. — URL: https://www.typescriptlang.org/docs/ (дата обращения: 16.09.2023).
2. React Tutorial. — Текст : электронный // W3Schools : [сайт]. — URL: https://www.w3schools.com/react/ (дата обращения: 17.09.2023).
3. Learn React. — Текст : электронный // React : [сайт]. — URL: https://react.dev/learn (дата обращения: 17.09.2023).
4. Prisma Docs. — Текст : электронный // Prisma : [сайт]. — URL: https://www.prisma.io/docs (дата обращения: 20.09.2023).
5. SASS Documentation. — Текст : электронный // SASS : [сайт]. — URL: https://sass-lang.com/documentation/ (дата обращения: 22.09.2023).
6. Documentation. — Текст : электронный // Docker docs : [сайт]. — URL: https://docs.docker.com/ (дата обращения: 10.10.2023).
7. Docker Compose overview. — Текст : электронный // Docker docs : [сайт]. — URL: https://docs.docker.com/compose/ (дата обращения: 11.10.2023).
8. Черный, Б. Профессиональный TypeScript. Разработка масштабируемых JavaScript приложений / Б. Черный. — 1-е изд. — Санкт-Петербург : Питер, 2021. — 352 c. — Текст : непосредственный.
9. Обзор API Gmail. — Текст : электронный // Gmail Workspace : [сайт]. — URL: https://developers.google.com/gmail/api/guides?hl=ru (дата обращения: 03.11.2023).
10. Documentation. — Текст : электронный // Socket IO : [сайт]. — URL: https://socket.io/docs/v4/ (дата обращения: 20.10.2023).

# **Приложение А**

Техническое задание

# **ПРИЛОЖЕНИЕ Б**

Технический проект

# **ПРИЛОЖЕНИЕ В**

Руководство программиста

# **ПРИЛОЖЕНИЕ Г**

Руководство пользователя

# **ПРИЛОЖЕНИЕ Д**

ПМИ

# **ПРИЛОЖЕНИЕ Е**

**Исходный код**

Исходный код клиентской части:

import { useContext, useRef, useEffect, useState } from "react";

import { useNavigate } from "react-router-dom";

import { AxiosError } from "axios";

import axios from "../tools/api";

import "../assets/styles/board.scss";

import ModalCreate from "./ModalCreate";

import ModalEdit from "./ModalEdit";

import { Task, Category } from "./Interfaces";

import { SocketContext } from "../context/socket";

import pencil from "../assets/img/pencil.svg";

import { dragOverHandler, dragStartHandler, dropHandler } from "../tools/dragndrop";

interface category {

id: number;

name: string;

}

interface MyError extends Error {

data: any;

}

function BoardPage() {

const { socket } = useContext(SocketContext);

const navigate = useNavigate();

const catInputRef = useRef<HTMLInputElement>(null);

const taskInputRef = useRef<HTMLInputElement>(null);

const descriptionInputRef = useRef<HTMLTextAreaElement>(null);

const [categories, setCategories] = useState<category[]>([]);

const [tasks, setTasks] = useState<Task[]>([]);

const [categoryId, setCatId] = useState(0);

const [openCreateCategory, setOpenCreateCategory] = useState(false);

const [openCreateTask, setOpenCreateTask] = useState(false);

const [openModalEdit, setOpenModalEdit] = useState(false);

const [editingItem, setEditingItem] = useState<Task | Category>();

const catDialogRef = useRef<HTMLDialogElement>(null);

const taskDialogRef = useRef<HTMLDialogElement>(null);

const [, setError] = useState("");

const [, setCurrentCard] = useState<Category>();

const [currentTaskCard, setCurrentTaskCard] = useState<Task>();

useEffect(() => {

const onConnect = () => {

socket.emit("get\_categories");

socket.emit("get\_tasks");

};

const onDisconnect = () => {};

const onError = (error: Error) => {

console.log("Error:", error.message);

console.log((error as MyError).data);

if ((error as MyError).data) setError((error as MyError).data.type);

};

const onTasks = (tasks: Task[]) => {

setTasks(tasks);

};

const onNewTask = (task: Task) => {

if (tasks.indexOf(task) === -1) setTasks((previous) => [...previous, task]);

};

const onEditTask = (task: Task) => {

setTasks((previous) => {

const taskIndex = previous.findIndex((item) => item.id === task.id);

const editedTasks = previous;

editedTasks[taskIndex] = task;

return [...editedTasks];

});

};

const onDeleteTask = (task: Task) => {

setTasks((previous) => [...previous.filter((item) => item.id !== task.id)]);

};

const onCategory = (categories: Category[]) => {

setCategories(categories);

};

const onNewCategory = (category: Category) => {

if (categories.indexOf(category) === -1) setCategories((previous) => [...previous, category]);

};

const onEditCategory = (category: Category) => {

setCategories((previous) => {

const categoryIndex = previous.findIndex((item) => item.id === category.id);

const editedCategories = previous;

editedCategories[categoryIndex] = category;

return [...editedCategories];

});

};

const onDeleteCategory = (category: Category) => {

setCategories((previous) => [...previous.filter((item) => item.id !== category.id)]);

};

socket.on("connect", onConnect);

socket.on("disconnect", onDisconnect);

socket.on("connect\_error", onError);

socket.on("tasks", onTasks);

socket.on("new\_task", onNewTask);

socket.on("edit\_task", onEditTask);

socket.on("move\_task", onEditTask);

socket.on("del\_task", onDeleteTask);

socket.on("categories", onCategory);

socket.on("new\_category", onNewCategory);

socket.on("edit\_category", onEditCategory);

socket.on("del\_category", onDeleteCategory);

socket.on("error", onError);

return () => {

socket.off("connect", onConnect);

socket.off("disconnect", onDisconnect);

socket.off("connect\_error", onError);

socket.off("tasks", onTasks);

socket.off("new\_task", onNewTask);

socket.off("edit\_task", onEditTask);

socket.off("move\_task", onEditTask);

socket.off("del\_task", onDeleteTask);

socket.off("categories", onCategory);

socket.off("new\_category", onNewCategory);

socket.off("edit\_category", onEditCategory);

socket.off("del\_category", onDeleteCategory);

socket.off("error", onError);

};

}, [socket, categories, tasks]);

useEffect(() => {

socket.connect();

// eslint-disable-next-line react-hooks/exhaustive-deps

}, []);

async function createCategory() {

const categoryName = catInputRef.current?.value || "";

const pass = localStorage.getItem("Password") || "";

try {

await axios.post(

"/category/new",

{

name: categoryName,

},

{

headers: { Authorization: pass },

},

);

if (catDialogRef.current) catDialogRef.current.style.visibility = "hidden";

} catch (error) {

if ((error as AxiosError).response) {

console.error("Error", error);

}

}

}

async function createTask(catId: number) {

const taskName = taskInputRef.current?.value || "";

const description = descriptionInputRef.current?.value || "";

const pass = localStorage.getItem("Password") || "";

try {

await axios.post(

"/task/new",

{

name: taskName,

category\_id: catId,

description: description,

},

{

headers: { Authorization: pass },

},

);

if (taskDialogRef.current) taskDialogRef.current.style.visibility = "hidden";

} catch (error) {

if ((error as AxiosError).response) {

console.error("Error", error);

}

}

}

async function editItem(item: Category | Task) {

const pass = localStorage.getItem("Password") || "";

const url = "/" + ("categoryId" in item ? "task" : "category") + "/edit";

try {

await axios.post(url, item, {

headers: { Authorization: pass },

});

} catch (error) {

if ((error as AxiosError).response) {

console.error("Error", error);

}

}

}

async function deleteItem(item: Category | Task) {

const pass = localStorage.getItem("Password") || "";

const url = "/" + ("categoryId" in item ? "task" : "category") + "/delete";

try {

await axios.post(url, item, {

headers: { Authorization: pass },

});

} catch (error) {

if ((error as AxiosError).response) {

console.error("Error", error);

}

}

}

return (

<div className='home-page'>

<div className='menu'>

<h1 className='menu\_\_header'>Board</h1>

<div className='menu\_\_btns'>

<button

className='menu\_\_btn'

onClick={() => {

navigate("/", { replace: true });

}}

>

Go back

</button>

<button

className='menu\_\_btn'

onClick={() => {

localStorage.removeItem("Password");

navigate("/login", { replace: true });

}}

>

Log Out

</button>

</div>

</div>

{openCreateCategory ? (

<ModalCreate

name='Category'

setOpen={setOpenCreateCategory}

onConfirm={createCategory}

nameInputRef={catInputRef}

descriptionInputRef={null}

/>

) : null}

{openCreateTask ? (

<ModalCreate

name='Task'

setOpen={setOpenCreateTask}

onConfirm={() => {

createTask(categoryId);

}}

nameInputRef={taskInputRef}

descriptionInputRef={descriptionInputRef}

/>

) : null}

{openModalEdit && editingItem ? (

<ModalEdit

item={editingItem}

setOpen={setOpenModalEdit}

onConfirm={editItem}

onDelete={() => {

if (window.confirm("Are you sure?")) {

deleteItem(editingItem);

setOpenModalEdit(false);

}

}}

/>

) : null}

<button

id='new-cat-btn'

onClick={() => {

setOpenCreateCategory(true);

}}

>

<h2>New</h2>

</button>

<div className='desk'>

{categories &&

categories.map((item) => {

// const cardName = item.name.length > 15 ? item.name.substring(0, 15) + "..." : item.name;

return (

<div

key={item.id}

className='category'

onDragOver={(e) => dragOverHandler(e)}

onDrop={(e) => dropHandler(e, item, currentTaskCard)}

>

<div className='category\_\_header'>

<h2>{item.name} </h2>

<button

className='category\_\_edit-btn'

onClick={() => {

setOpenModalEdit(true);

setEditingItem(item);

}}

>

<img src={pencil} className='category\_\_edit-btn-img' alt='edit' />

</button>

</div>

<button

className='category\_\_create-task-btn'

onClick={() => {

setCatId(item.id);

setOpenCreateTask(true);

}}

>

+

</button>

<div className='tasks'>

{tasks

.filter((task) => {

return task.categoryId === item.id;

})

.map((task) => {

// const taskName =

// task.name.length > 23 ? task.name.substring(0, 23) + "..." : task.name;

return (

<div

className='tasks\_\_task'

key={task.id}

onDragStart={(e) =>

dragStartHandler(

e,

item,

task,

setCurrentCard,

setCurrentTaskCard,

)

}

onDrop={(e) => dropHandler(e, item, currentTaskCard)}

draggable={true}

>

<div>{task.name}</div>

{/\* TODO: Add description indicator

{task.description ? <div className='desc'>desc</div> : ""} \*/}

<button

className='tasks\_\_task-edit-btn'

onClick={() => {

setOpenModalEdit(true);

setEditingItem(task);

}}

>

<img src={pencil} alt='edit' />

</button>

</div>

);

})}

</div>

</div>

);

})}

</div>

</div>

);

}

export default BoardPage;

**Исходный код серверной части:**

import express from "express";

import { json, urlencoded } from "body-parser";

// import multer from "multer";

import user from "./views/user";

import category from "./views/category";

import task from "./views/task";

import morgan from "morgan";

import cors from "cors";

import http from "http";

import { Server } from "socket.io";

import { PrismaClient } from "@prisma/client";

import { userMiddleware } from "./views/tools/middleware";

// const storage: multer.StorageEngine = multer.diskStorage({

// destination: (\_req, \_file, \_cb) => {

// \_cb(null, "uploads");

// },

// filename: (\_req, \_file, \_cb) => {

// \_cb(null, Date.now() + "-" + \_file.originalname);

// },

// });

const prisma = new PrismaClient();

const app: express.Application = express();

// const upload: multer.Multer = multer({ storage: storage });

const server = http.createServer(app);

const io = new Server(server, {

cors: {

origin: "\*",

},

// transports: ["polling"],

// pingTimeout: 180000,

});

const port: number = 3500;

app.use(json());

app.use(urlencoded({ extended: true }));

app.use(express.static("public"));

app.use(cors());

app.use(morgan("dev"));

app.set("Access-Control-Allow-Origin", "\*");

app.get("/", (\_req, \_res) => {

\_res.status(200).send({ status: "ok" });

});

app.use("/user", user);

app.use(userMiddleware);

app.use("/category", category(io));

app.use("/task", task(io));

// app.post("/profile", upload.single("avatar"), (\_req, \_res, \_next) => {

// console.log(\_req.file, \_req.body);

// \_res.json(\_req.body);

// });

interface MyError extends Error {

data: any;

}

io.use(async (socket, next) => {

const pass = socket.handshake.auth.pass;

if (pass === process.env.PASSWORD) {

return next();

}

const err = new Error("bad pass") as MyError;

err.data = { type: "Wrong pass" };

return await next(err);

});

io.on("connection", (socket) => {

socket.on("disconnect", () => {

socket.removeAllListeners();

});

socket.on("get\_tasks", async () => {

const tasks = await prisma.task.findMany();

socket.emit("tasks", tasks);

});

socket.on("get\_categories", async () => {

const categories = await prisma.category.findMany();

socket.emit("categories", categories);

});

});

server.listen(port, () => {

console.log(`TypeScript with Express on http://localhost:${port}/`);

});

**Конфигурация сборки Docker Compose:**

version: '3.3'

services:

frontend:

build:

context: ./frontend

image: mrsedan/neuro-trello-frontend

container\_name: frontend\_neuro\_trello

networks:

- labnet

ports:

- 80:80

depends\_on:

- backend

restart: unless-stopped

backend:

build: ./backend

image: mrsedan/neuro-trello-backend

container\_name: backend\_neuro\_trello

environment:

- CHOKIDAR\_USEPOLLING=true

volumes:

- ./backend/prisma:/app/prisma

- ./backend/.env:/app/.env

networks:

- labnet

ports:

- 3500:3500

networks:

labnet: