МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

**«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(МОСКОВСКИЙ ПОЛИТЕХ)**

09.03.01 Информатика и вычислительная техника

Образовательные программы (профили)

«Интеграция и программирование в САПР»,

«Веб-технологии"

ОТЧЕТ

по дисциплине:

**Проектная деятельность**

на тему:

Создание мобильного приложения на платформе VK-Mini-Apps

Преподаватель: / Холодов Д. А. /

*подпись ФИО, уч. звание и степень*

Студент: / Панаскин А. Д., 201-323 /

*подпись ФИО, группа*

Студент: / Сергеев О. П., 201-323 /

*подпись ФИО, группа*

Студент: / Воробьёв. В. В., 201-321 /

*подпись ФИО, группа*

Студент: / Бодур Э. А. М, 201-321 /

*подпись ФИО, группа*

Москва, 2021 г.

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc75726338)

[1 ОБЩИЕ СВЕДЕНИЯ О ПРОЕКТЕ 4](#_Toc75726339)

[1.1 Актуальность проекта 4](#_Toc75726340)

[1.2 Цели проекта 4](#_Toc75726341)

[2 ОБЩЕЕ ЗАДАНИЕ НА ПРОЕКТ 5](#_Toc75726342)

[3 ЗАДАЧИ ПРОЕКТА 6](#_Toc75726343)

[12. ПЛАН РАБОТЫ 7](#_Toc75726344)

[13. ИНДИВИДУАЛЬНЫЕ ПЛАНЫ УЧАСТНИКОВ 8](#_Toc75726345)

[5.1 Выбор технологий для реализации 10](#_Toc75726346)

[5.2 Распределение задач между участниками и командная работа 10](#_Toc75726347)

[5.3 Описание структуры приложения 10](#_Toc75726348)

[5.4 Программное содержимое 10](#_Toc75726349)

[5.5 Описание структуры данных 15](#_Toc75726350)

[5.5.1 Сбор информации для раздела «Частые вопросы» 16](#_Toc75726351)

[5.6 Инструмент заполнения данных 18](#_Toc75726352)

[5.7 Тестирование приложения 19](#_Toc75726353)

[5. 8 Написание руководства пользователя и инструкции к развёртке 20](#_Toc75726354)

[14. РЕЗУЛЬТАТЫ 21](#_Toc75726355)

[14.1 Достигнутые цели 21](#_Toc75726356)

[14.2 Материалы и ссылки проекта 21](#_Toc75726357)

# **ВВЕДЕНИЕ**

Для работы по дисциплине «Проектная деятельность» команда выдвинула инициативный проект: «Приложение-справочник для абитуриентов и первокурсников факультета информационных технологий». Предметной областью выбранного нами проекта является изучение создание приложений на основе платформы VK-Mini-Apps. Данный проект основывается на технологии интеграции пользовательских приложений в мобильный и веб-клиент социальной сети «ВКонтакте». Наша команда изучила материалы, связанные с разработкой, развёрткой и поддержкой приложений, чтобы в итоге представить работающее приложение и инструкцию по развёртыванию своего проекта на данной платформе.

# **ОБЩИЕ СВЕДЕНИЯ О ПРОЕКТЕ**

## **Актуальность проекта**

В данный момент социальная сеть ВКонтакте остаётся самой популярной социальной сетью в России и на территории СНГ. При помощи нашего приложения абитуриенты смогут узнать про направления факультета информационных технологий, а студенты найти решения распространённых вопросов, не выходя из социальной сети и делится приложением с друзьями и одногруппниками.

## **Цели проекта**

Целью нашего проекта является разработка кроссплатформенного приложения и составление инструкции по развёртке собственного приложения.

# **ОБЩЕЕ ЗАДАНИЕ НА ПРОЕКТ**

Основная цель проекта заключалась в изучении библиотек и методов, предоставленных разработчиками ВКонтакте и разработки на основе полученных знаний. Также необходимо составить инструкцию по созданию пользовательских приложений на основе командного.

# **ЗАДАЧИ ПРОЕКТА**

1. Продумать логики приложения.
2. Создать макет приложения исходя из логики (1).
3. Изучение библиотек и методов разработки для выбранной платформы.
4. Изучение принципов взаимодействия модулей приложения друг с другом.
5. Выбор типа хранения не статичных данных в приложении.
6. Создание эскизных моделей приложения.
7. Исправление программных неисправностей.
8. Создание финальной модели приложения.
9. Наполнение приложения контентом.
10. Создание сопроводительных материалов – страницу в социальной сети, постер, руководство пользователя, инструкцию по развёртке проекта.
11. Создание отчётности по проекту.

# **ПЛАН РАБОТЫ**

1. Определение основных целей проекта.
2. Постановка задач.
3. Введение в суть проекта участников.
4. Распределение заданий.
5. Создание первоначальных модулей приложения и их объединение.
6. Исправление багов и корректировка модулей.
7. Создание сопроводительных материалов.
8. Подготовка отчетной документации.

# **ИНДИВИДУАЛЬНЫЕ ПЛАНЫ УЧАСТНИКОВ**

|  |  |  |
| --- | --- | --- |
| Участник проекта | Группа | Индивидуальный план |
| Панаскин А. Д. | 201-323 | 1. Определение основных целей проекта. 2. Постановка и распределение задач. 3. Обсуждение и выбор решений и методов. 4. Проектирование логики приложения. 5. Создание и наполнение JSON-файлов. 6. Наполнение приложения контентом. 7. Подготовка постера, Github репозитория, итогового отчёта и лэндинга проекта. |
| Воробьёв В. В. | 201-321 | 1. Обсуждение и выбор решений и методов. 2. Поиск подходящих решений и компонентов. 3. Написание кода модулей приложения. 4. Объединение модулей других участников. 5. Тестирование и отладка. 6. Рефакторинг кода. 7. Создание макета приложения в Figma. |
| Сергеев О. П. | 201-323 | 1. Обсуждение и выбор решений и методов. 2. Поиск подходящих решений и компонентов. 3. Исследование сторонних ресурсов для интеграции в приложение. 4. Объяснение принципов работы отдельных компонентов. 5. Создание инструмента наполнения JSON-файлов; 6. Тестирование приложения. |
| Бодур Э. А. М. | 201-321 | 1. Обсуждение и выбор решений и методов. 2. Поиск подходящих решений и компонентов. 3. Написание кода модулей приложения. 4. Рефакторинг кода. 5. Подборка дизайнерских решений. 6. Работа с внешним видом приложения. 7. Иллюстрация логики приложения. |

6 **СОДЕРЖАТЕЛЬНЫЕ ГЛАВЫ**

## **6.1 Выбор технологий для реализации**

Для выбора технологий написания приложения было проведено командное обсуждение. В ходе обсуждения было решено использование React-JS для кода приложения и JSON-файлы для хранения данных про направления, общежития и частые вопросы.

## **6.2 Распределение задач между участниками и командная работа**

Для разработки проекта команда выбрала модульный принцип. Каждый участник получал задачи по распределению и работал только с ними. После истечения срока задач команда собиралась на обсуждение и объединение результатов. Если текущий вариант приложения всех устраивал – задачи обновлялись и цикл повторялся. Если текущий вариант не удовлетворял ожиданиям команды – разногласия решались в рамках текущего сбора.

На финальных этапах проекта, задачи перестали прикрепляется к конкретному участнику команды и выполнялись из общего бассейна задач.

Распределение, выставление сроков и выполнение задач велось в онлайн-менеджере задач Trello – <https://trello.com/b/rDzb3CWw/polyapp>.

## **6.3 Описание структуры приложения**

Приложение рассчитано на абитуриентов и студентов, поэтому логика приложения делится на две ветки. Общая схема логики изображена в приложении 1. Экраны приложения изображены в приложении 2.

## **6.4 Программное содержимое**

Программный код был реализован на языке React-Js, именно на нём и написаны компоненты из VK-UI. На статических страницах код заключается в правильном расположении компонентов относительно экрана пользователя и других компонентов. На страницах, на которых содержимое должно изменятся относительно выбора пользователя, было необходимо реализовать условный рендеринг компонентов. Другими словами, проверку какие условия выбрал пользователь на предыдущем экране. Пример выставления параметров на основе выбора пользователя со стартового экрана можно изучить ниже.

return (

        <Panel id={id}>

            <PanelHeader>PolyApp</PanelHeader>

            {fetchedUser &&

                <Group header={<Header mode="primary">Привет!</Header>}>

                    <Div className="homepage-subhead">

                        <Subhead weight="semibold">Это приложение для абитуриентов и студентов факультета информационных технологий Московского политеха.</Subhead>

                    </Div>

                    <Cell

                        before={fetchedUser.photo\_200 ? <Avatar src={fetchedUser.photo\_200} /> : null}

                        description={fetchedUser.city && fetchedUser.city.title ? fetchedUser.city.title : ''}

                    >

                        {`${fetchedUser.first\_name} ${fetchedUser.last\_name}`}

                    </Cell>

                </Group>}

            <Group>

                <Div>

                    <FormLayout>

                        <FormItem top="Расскажи про себя">

                            <Radio name="radio" onClick={() => setWant(false)} value="already" defaultChecked>Я уже студент</Radio>

                            <Radio name="radio" onClick={() => scrollToSection()} value="newStudent">Я хочу поступить</Radio>

                        </FormItem>

                    </FormLayout>

                </Div>

                {want ?

                    <Div style={{ marginBlockEnd: 50 }}>

                        <Caption className="captionCaps" level="1" weight="semibold" caps >

                            Круто, хороший выбор!

                        </Caption>

                        <FormLayout>

                            <Header mode="primary" aside={<Icon20QuestionOutline onClick={() => setActiveModal(MODAL\_CARD\_ONE)} />} style={{ marginBottom: 10 }}>Форма обучения</Header>

                            <FormItem><Radio name="radio" value="Очная" onClick={() => setForm('Очная')} defaultChecked>Очная</Radio>

                                <Radio name="radio" value="Заочная" onClick={() => setForm('Заочная')}>Заочная</Radio></FormItem>

                        </FormLayout>

                        {wantStudyForm == 'Очная' ?

                            <FormLayout>

                                <Header mode="primary" style={{ marginBottom: 10 }} aside={<Icon20QuestionOutline onClick={() => setActiveModal(MODAL\_CARD\_TWO)} />}>Ступень образования</Header>

                                <FormItem><Radio name="radio" value="Бакалавриат" onClick={() => setLevel('Бакалавриат')} defaultChecked>Бакалавриат</Radio>

                                    <Radio name="radio" value="Специалитет" onClick={() => setLevel('Специалитет')} >Специалитет</Radio>

                                    <Radio name="radio" value="Магистратура" onClick={() => setLevel('Магистратура')} >Магистратура</Radio></FormItem>

                            </FormLayout>

                            :

                            <FormLayout>

                                <Header mode="primary" style={{ marginBottom: 10 }} aside={<Icon20QuestionOutline onClick={() => setActiveModal(MODAL\_CARD\_TWO)} />}>Ступень образования</Header>

                                    <FormItem><Radio name="radio" value="Бакалавриат" onClick={() => setLevel('Бакалавриат')} defaultChecked>Бакалавриат</Radio></FormItem>

                            </FormLayout>

                        }

                    </Div>

                    : null

                }

                <FixedLayout filled vertical="bottom">

                    <Div>

                        <Button stretched size="l" mode="primary" onClick={(event) => handleNextClick(event)} data-to={want ? 'pick-directions' : 'student-form-filling'} >

                            Далее

                        </Button>

                    </Div>

                </FixedLayout>

            </Group>

        </Panel>

    );

Также необходимо было реализовать правильный рендеринг записей из JSON-файлов. Рассмотрим на примере файла «Направления». Алгоритм должен был понимать какую форму обучения и уровень подготовки выбрал пользователь. После, вывести из соответствующей ветки JSON-файла подходящие данные и ждать выбора направлений от пользователя. При нажатии на значок «Инфо», происходит переход на страницу конкретного направления. Для этого экрана необходимо собирать данные уже по направлению и переносить поля из файла (код, количество мест, учебный план и т. д.) в соответствующие поля приложения. Схема переноса данных из JSON-файла в приложение показана на рисунке 1.

Рисунок 1 - Иллюстрация переноса данных из JSON-файла в приложение

Ниже вы можете изучить код рендеринга из схемы выше. Весь код можно изучить в приложении 3.

class AboutDirection extends Component {

    constructor(props) {

        super(props);

        this.state = {

            interestedDirection: ''

        }

    }

    componentDidMount = () => {

        this.setState({ interestedDirection: JSON.parse(localStorage.getItem('interestedDirection')) })

    }

    createMarkup = (text) => {

        return { \_\_html: text };

      }

    render() {

        return (

            <Panel id={this.props.id}>

                <PanelHeader left={<PanelHeaderBack onClick={this.props.back} />}>PolyApp</PanelHeader>

                <Title style={{ marginLeft: 16, marginRight: 16, marginTop: 16, padding: 16, background: 'var(--content\_tint\_background)', borderRadius: 7, fontSize: '1.2rem', textAlign: 'center' }}>

                    {this.state.interestedDirection['Название направления']}

                </Title>

                <Title level='2' weight="regular" style={{ marginLeft: 16, marginRight: 16, marginTop: 16, padding: '16 0 16 0', borderRadius: 7, textAlign: 'left', color: 'var(--text-primary)' }}>

                    о направлении:

        </Title>

                <List>

                    <Cell indicator={this.state.interestedDirection['Код направления']}>Код</Cell>

                    <Cell indicator={this.state.interestedDirection['Срок обучения']}>Срок обучения</Cell>

                    <Cell indicator={this.state.interestedDirection['Бюджетных мест']}>Бюджетных мест</Cell>

                    <Cell indicator={this.state.interestedDirection['Платных мест']}>Платных мест</Cell>

                </List>

                <Text weight="medium" style={{ marginBottom: 16, padding: 16, marginBlockEnd: 70}} dangerouslySetInnerHTML={this.createMarkup(this.state.interestedDirection['Описание'])}></Text>

                <FixedLayout filled vertical="bottom">

                <Div style={{ display: 'flex', flexDirection: 'row', alignItems: 'center', justifyContent: 'center!important' }}>

                    <Link style={{width: '100%', textAlign: 'center', color:"var(--button\_primary\_foreground)",

                    background:"var(--button\_primary\_background)", borderRadius:8, minHeight:44, display: "flex", justifyContent: "center", alignItems: "center", fontWeight:"500"}}

                    href={this.state.interestedDirection['Подробнее']} target="\_blank"> Подробнее</Link>

                    </Div>

                </FixedLayout>

            </Panel>

        )

    };

## **6.5 Описание структуры данных**

Для хранения не статических данных, таких как «Направления», «Общежития» и «Частые вопросы» используются JSON-файлы с различной структурой вложенности. Например, для файла «Направления» стартовым разветвлением выступает форма обучения, целиком структура выглядит следующим образом:

* Очная
  + Бакалавриат
    - Информатика и вычислительная техника - 09.03.01
      * Веб-технологии
        + \*Внутренние поля каждого направления\*
      * Интегрирование и программирование в САПР
      * Киберфизические системы
  + Специалитет
  + Магистратура
* Заочная

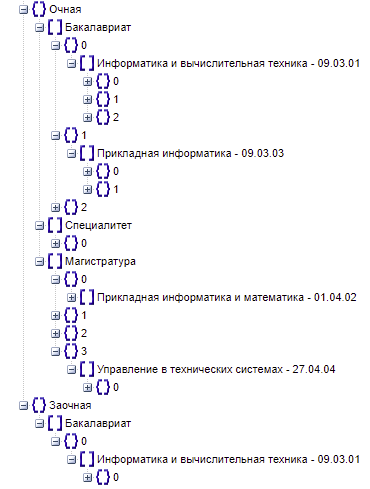
Структуру каждого из фалов можно изучить на рисунках 2, 3 и 4. Изучить содержимое JSON-файлов можно в приложении 4.

Рисунок 2 - Структура JSON - файла "Направления"

## **6.5.1 Сбор информации для раздела «Частые вопросы»**

Рисунок 3- Структура JSON-файла "Общежития"

Рисунок 4 - Структура JSON-файла "Вопросы"

Данный раздел был разделён на четыре основные категории: «Общежития», «Учёба», «Корпуса», «Проектная деятельность».

Чтобы приложение было полезным всем студентам было необходимо собрать наиболее распространённые вопросы студентов факультета информационных технологий. Для этого была создана гугл-форма и распространена по беседам первых курсов – <https://forms.gle/YJ6cVkdZC7P66QfY9>, рисунок 5.

Ниже приведены наиболее частые вопросы из ответов формы:

Рисунок 5 - Форма для вопросов от студентов

Рисунок 6 - Форма для вопросов от студентов

* Как добираться от одного корпуса к другому?
* Как правильно оформить переезд в общежитии?
* Как узнать даты пересдач?
* Как взять академический отпуск?

Пример ответа на вопрос «Как добраться от БС до ПК» показан на рисунке 6. При переходе по ссылке, открывается гугл-карта с проложенным маршрутом.

## **6.6 Инструмент заполнения данных**

Рисунок 6 - Пример ответа из категории "Корпуса"

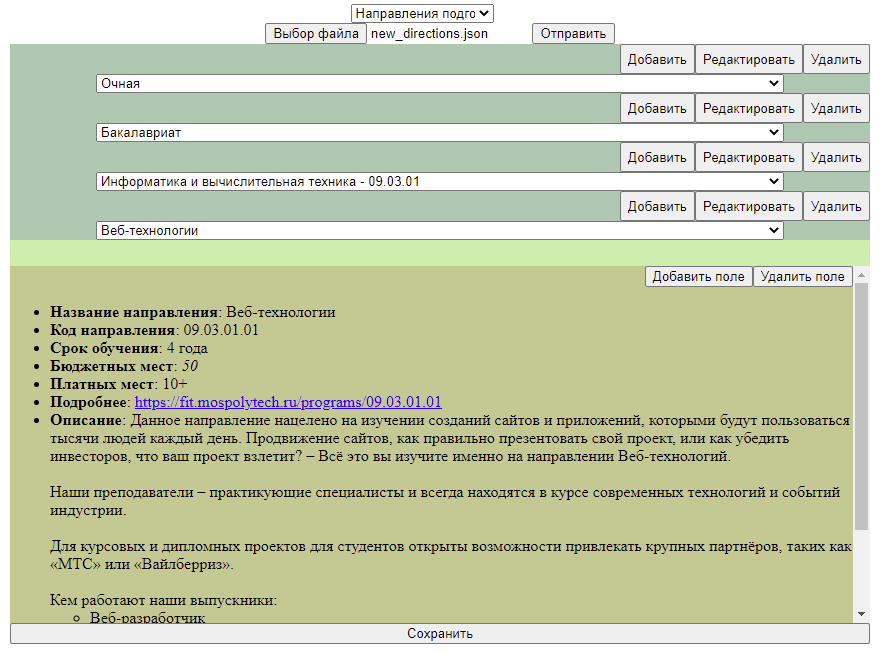
Для удобного заполнения файлов был разработан веб-инструмент для добавления, редактирования и удаления данных. Инструмент представляет собой веб-интерфейс с полями, совпадающими с полями в JSON-файлах. Инструмент рассчитан на локальное использование администраторами приложения и не планируется распространятся как самостоятельное приложение. Интерфейс показан на рисунке 7.

Рисунок 7 - Интерфейс веб-инструмента для заполнения файлов

## **6.7 Тестирование приложения**

В ходе разработки в приложении появлялись различные функциональные, структурные, логические и орфографические ошибки.

Так как приложение состоит из собранных во едино модулей-экранов, наиболее предпочтительным был модульный принцип тестирования. Участники команды по окончанию своего задания, сразу тестировали его на локальном приложении.

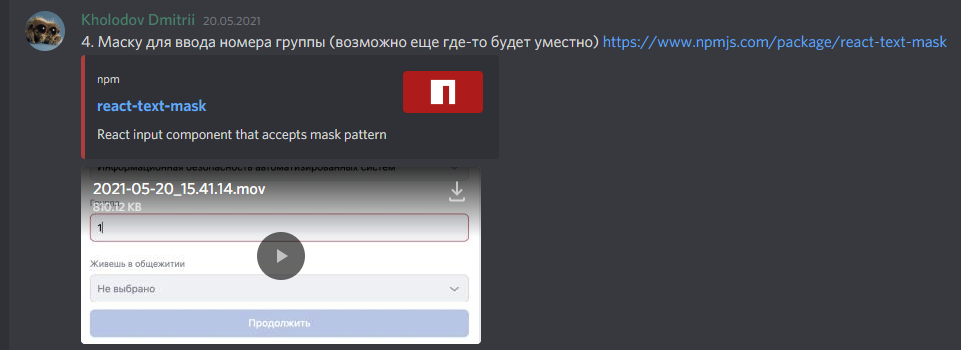
Функциональные недочёты помогал выявить и исправить куратор проекта. Пример замечания на рисунке 8.

Рисунок 8 - Помощь в исправлении бага от куратора

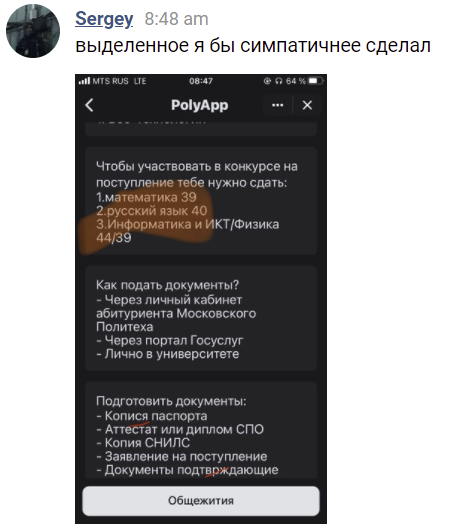
Орфографические ошибки помогала выявлять тестовая группа из знакомых и студентов первого курса. Пример на рисунке 9.

Рисунок 9 - Обратная связь от тестеров приложения

## **6. 8 Написание руководства пользователя и инструкции к развёртке**

В ходе работы, участники команды составляли рекомендации и советы как для простых пользователей приложения – абитуриентов и студентов, так и для разработчиков – начинающих программистов на React-JS.

Руководство пользователя представляет собой инструкцию по последовательным переходам между экранами приложения. Например, инструкция для абитуриентов выглядит следующим образом:

* 1. Нажмите по пункту «я хочу поступить» и ступень образования и форму обучения.
  2. Выберите интересующие вас направления и узнайте информацию про них, нажатием по кнопке «инфо» в строке каждого направления.
  3. После перехода на следующую страницу вы можете изучить какие предметы нужно сдать для поступления по данным направлениям, перейти на сайты, где можно подать документы онлайн или же перейти на страницу «Общежития».
  4. На странице «Общежития» вы можете узнать у каких студентов есть право получить общежитие и изучить каждое общежитие.

Полное руководство пользователя и инструкцию по развёртке проекта находятся в приложении 5 и 6.

# **РЕЗУЛЬТАТЫ**

## **Достигнутые цели**

В результате выполнения проекта команда достигла поставленных целей, именно:

1. Команда изучила документацию и принципы разработки написания приложений для платформы Vk-Mini-Apps;
2. Реализовала полученные знания на практике, написав код приложения;
3. Используя технологии GitHub, приложение функционирует онлайн и доступно всем желающим;
4. Для пользователей создано руководство пользователя, а для разработчиков – инструкция по развёртке проекта.

Скриншоты экранов приложении можно изучить в приложении №

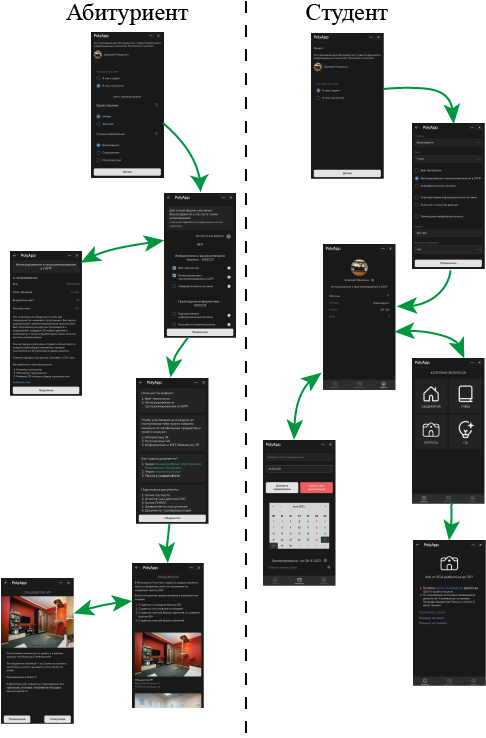
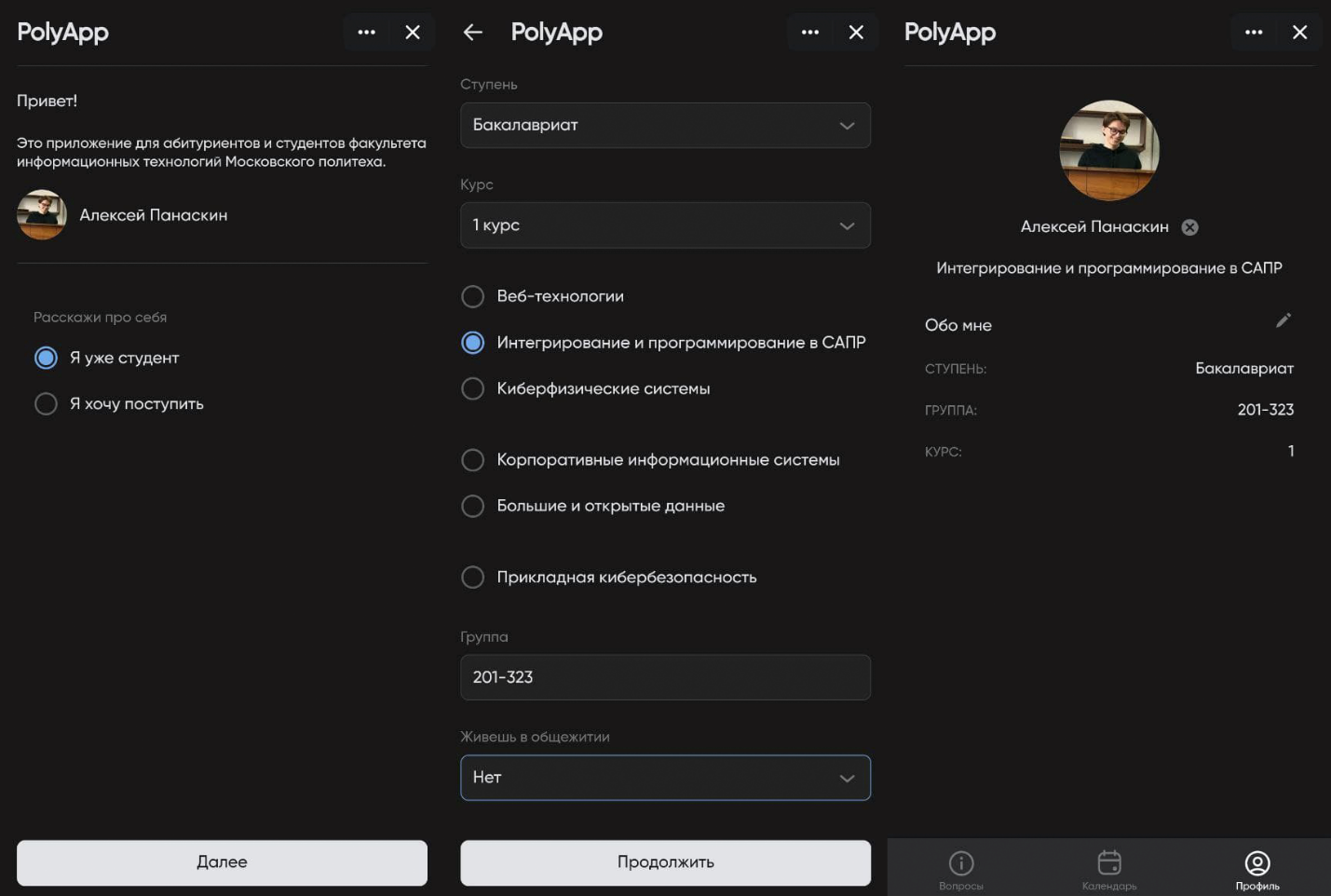
## **Материалы и ссылки проекта**

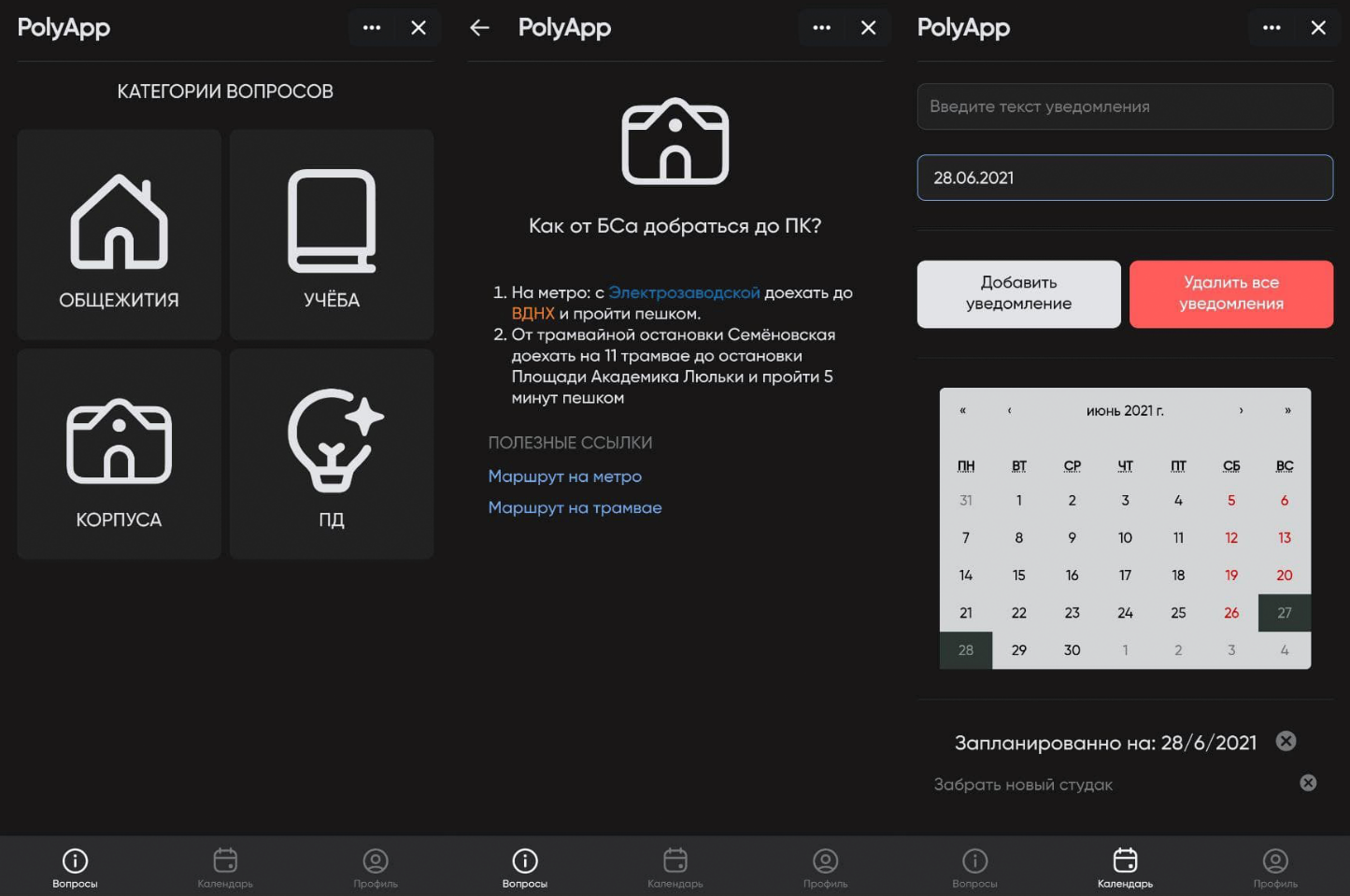
1. Приложение PolyApp – <https://vk.com/app7762604>
2. Страница в социальной сети ВКонтакте - <https://vk.com/polyapp_fit>
3. GitHub репозиторий проекта -<https://github.com/AlekseyPanaskin/PolyApp>
4. Лэндинг проекта – <http://polyapp.tilda.ws/>
5. Гугл-диск проекта – <https://drive.google.com/drive/folders/1GUbm8GQ-olZotfx8gfPIFIDIX9f50LLT>
6. Ресурс для распределения задач Trello – <https://trello.com/b/rDzb3CWw/polyapp>
7. Discord-сервер для командных обсуждений - <https://discord.gg/XVjAC5gt>

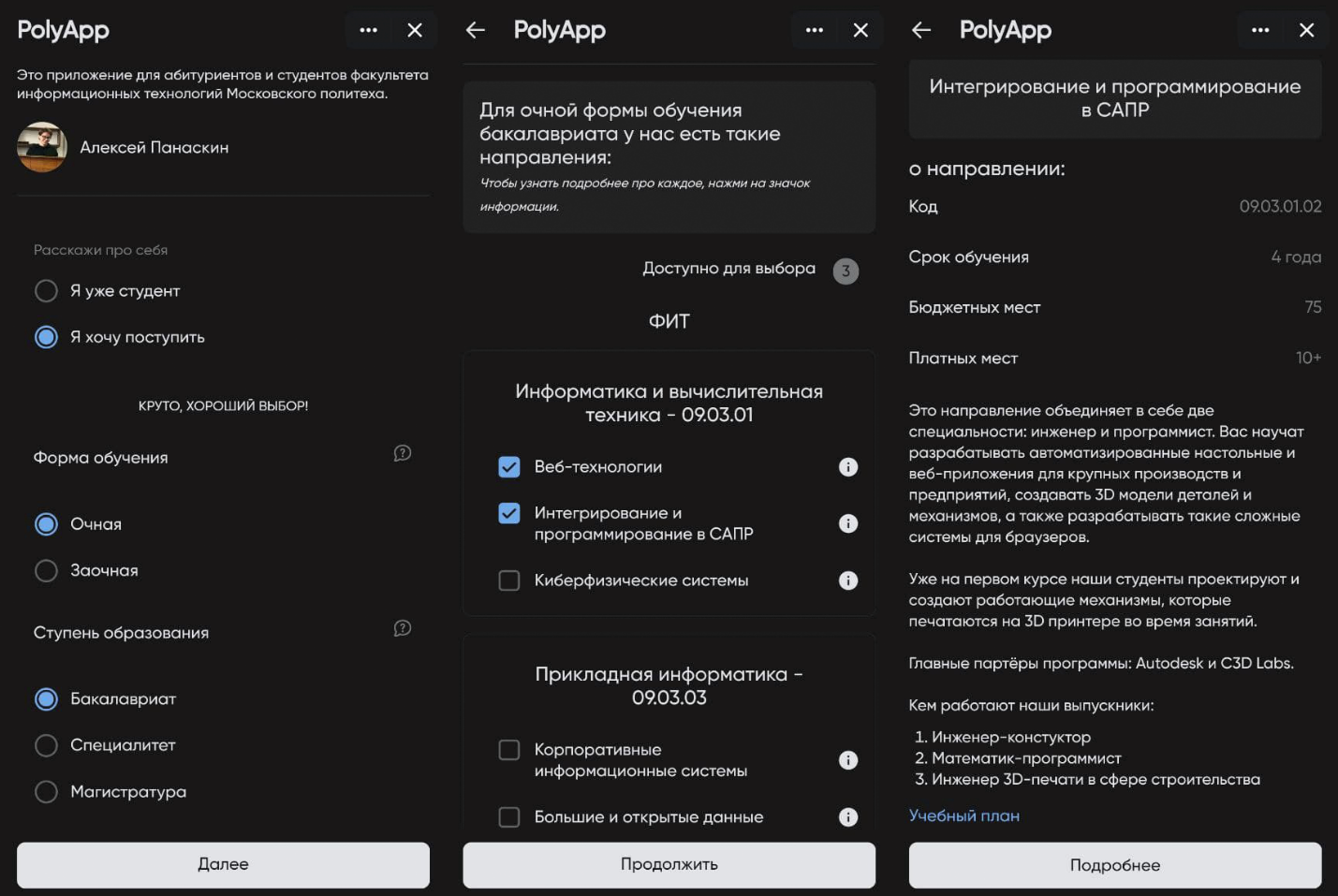
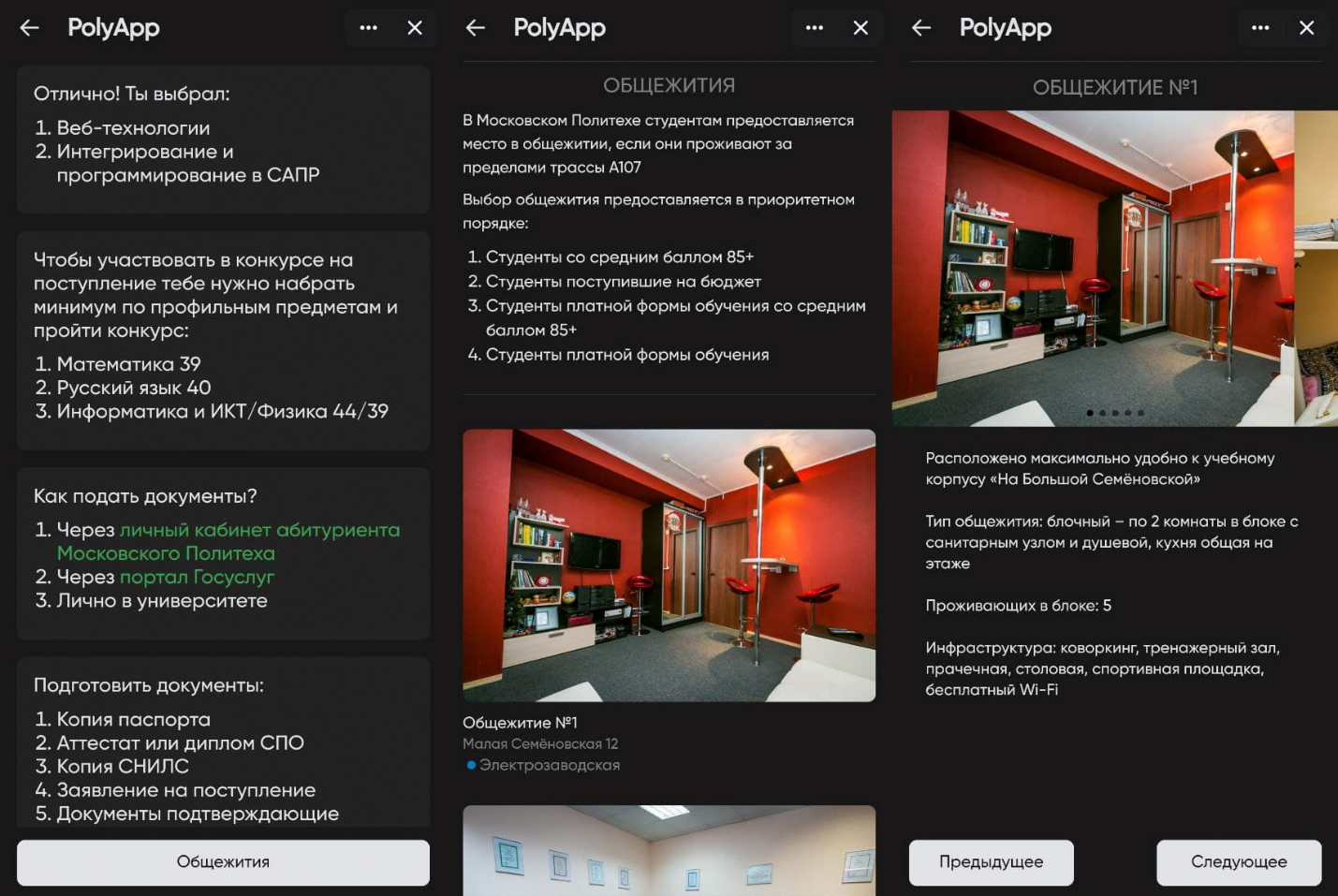
**СПИСОК ИСТОЧНИКОВ**

1. Команда разработчиков Вконтаке Быстрый старт в Vk-Mini-Apps [Электронный ресурс]. Сайт разработчиков. URL: <https://vk.com/dev>
2. Команда разработчиков Вконтаке VK-UI styleguide для Vk-mini-apps [Электронный ресурс]. Сайт разработчиков. URL: https://vkcom.github.io/VKUI/
3. Скот Чакон Основы работы с Git [Электронный ресурс]. Открытый интернет учебник. URL: <https://git-scm.com/book/ru/v2>

**ПРИЛОЖЕНИЯ**

1. Общая логика приложения
2. Экраны приложения





1. Листинг кода приложения

Файл «App.js»

import React, { useState, useEffect } from 'react';

import bridge from '@vkontakte/vk-bridge';

import View from '@vkontakte/vkui/dist/components/View/View';

import ScreenSpinner from '@vkontakte/vkui/dist/components/ScreenSpinner/ScreenSpinner';

import { AdaptivityProvider, AppRoot, ModalCard, ModalRoot, ModalPage, Text, ModalPageHeader, PanelHeaderButton,Div} from '@vkontakte/vkui';

import '@vkontakte/vkui/dist/vkui.css';

// import Start from './panels/Start';

import Acquaintance from './panels/Acquaintance';

// import StudyForm from './panels/StydyForm';

// import Degree from './panels/Degree';

import PickDirections from './panels/PickDirections\_v2';

import AboutDirection from './panels/AboutDirection';

import ChoosedDirectionsInfo from './panels/ChoosedDirectionsInfo';

import AboutStudent from './panels/AboutStudent';

import HomePage from './panels/HomePage';

import Questions from './panels/Questions';

import QuestionsList from './panels/QustionList';

import Instruction from './panels/Instruction';

import CalendarPanel from './panels/Calendar';

import Dorms from './panels/Dorms';

import DormPage from './panels/DormPage';

import EditStudent from './panels/EditStudent';

import './css/gilroy.css';

import './css/main.css';

import { Icon56QuestionOutline } from '@vkontakte/icons';

import { Icon56GhostOutline } from '@vkontakte/icons';

import arbuz from './img/arbuz.png'

const MODAL\_CARD\_ONE = 'modal-one';

const MODAL\_CARD\_TWO = 'modal-two';

const MODAL\_CARD\_THREE = 'modal-three';

const App = () => {

    const [activePanel, setActivePanel] = useState(localStorage.getItem('validation') ? 'home' : 'acquaintance');

    const [fetchedUser, setUser] = useState(null);

    const [popout, setPopout] = useState(<ScreenSpinner size='large' />);

    const [category, setCategory] = useState('');

    const [question, setQuestion] = useState(null);

    const [dorm, setdorm] = useState(0);

    const [activeModal, setActiveModal] = useState(null);

    function menu(e) {

        if (e.state) {

            setActivePanel(e.state.panel);

        } else {

            setActivePanel(routes.home);

        }

    }

    function back() { window.history.back(); }

    useEffect(() => {

        window.addEventListener('popstate', e => e.preventDefault() & menu(e));

        window.history.pushState({ panel: 'acquaintance' }, 'acquaintance');

        bridge.subscribe(({ detail: { type, data } }) => {

            if (type === 'VKWebAppUpdateConfig') {

                const schemeAttribute = document.createAttribute('scheme');

                schemeAttribute.value = data.scheme ? data.scheme : 'client\_light';

                document.body.attributes.setNamedItem(schemeAttribute);

            }

        });

        async function fetchData() {

            const user = await bridge.send('VKWebAppGetUserInfo');

            setUser(user);

            setPopout(null);

        }

        fetchData();

    }, []);

    const go = e => {

        setActivePanel(e.currentTarget.dataset.to);

        window.history.pushState({ panel: e.currentTarget.dataset.to }, `${e.currentTarget.dataset.to}`);

    };

    const updateData = (value) => {

        setCategory(value)

    }

    const updateQuestion = (value) => {

        setQuestion(value)

    }

    const modal = (

        <ModalRoot

            activeModal={activeModal}

        >

        <ModalCard

                id={MODAL\_CARD\_ONE}

                onClose={() => setActiveModal(null)}

                icon={<Icon56QuestionOutline />}

                header="Какие бывают формы обучения?"

            >

                      <Div>

            <Text>

            Очная - занятия проводятся по будням, и на некоторых направлениях по субботам. Если не посещать пары можно получить недопуск или даже отчислится.

            <br/><br/>Заочная - вы встречаетесь с преподавателями вживую только два раза в году - на сессии. Объем программы такой же как и на очной, но большую часть студент изучает самостоятельно.

            </Text>

          </Div>

            </ModalCard>

            <ModalCard

                id={MODAL\_CARD\_TWO}

                onClose={() => setActiveModal(null)}

                icon={<Icon56QuestionOutline />}

                header="Какие бывают ступени образования?"

            >

                                      <Div>

            <Text>

                Бакалавриат - базовый уровень высшего образования. Поступать на бакалавриат имеют право выпускники 11 класса, сдавшие егэ или выпускники колледжей по конкурсу внутренних экзаменов ВУЗа. Обучение длится 4 года.

                <br/><br/>Специалитет - тоже высшее образование, но выпускники получают конкретную профессию и возможность преподавать. Обучение длится 5 лет.

                <br/><br/>Магистратура - второй этап высшего образования.

                Для поступления необходимо иметь диплом бакалавра или специалитета. Обучение длится 2 года

            </Text>

          </Div>

            </ModalCard>

            <ModalCard

                id={MODAL\_CARD\_THREE}

                onClose={() => setActiveModal(null)}

                icon={<Icon56GhostOutline />}

                header="Просим прощения"

            >

                                      <Div>

            <Text>

                К сожалению, страница календаря ещё находится в разработке.

            </Text>

          </Div>

            </ModalCard>

        </ModalRoot>

    );

    return (

        <AdaptivityProvider>

            <AppRoot>

                <View activePanel={activePanel} popout={popout} modal={modal}>

                    {/\* <Start id='start' go={go} /> \*/}

                    <Acquaintance setActiveModal={setActiveModal} fetchedUser={fetchedUser}

                    MODAL\_CARD\_ONE={MODAL\_CARD\_ONE} MODAL\_CARD\_TWO={MODAL\_CARD\_TWO}

                    id='acquaintance' go={go} />

                    {/\* <StudyForm id='study-form' go={go} />

                    <Degree id='degree' go={go}/> \*/}

                    <PickDirections back={back} id='pick-directions' go={go} />

                    <AboutDirection back={back} id='about-direction' go={go} />

                    <ChoosedDirectionsInfo back={back} id='choosed-directions-info' go={go} />

                    <DormPage back={back} dorm={dorm} setdorm={setdorm} id='dorm-page' go={go} />

                    <Dorms back={back} setdorm={setdorm} id='dorms' go={go} choosedDorm={0} />

                    {/\* Ветка два \*/}

                    <AboutStudent back={back} id="student-form-filling" go={go} />

                    <EditStudent back={back} go={go} id="edit" setActivePanel={setActivePanel} />

                    <HomePage MODAL\_CARD\_THREE={MODAL\_CARD\_THREE} setActiveModal={setActiveModal}  id='home' setActivePanel={setActivePanel} fetchedUser={fetchedUser} go={go} />

                    <Questions MODAL\_CARD\_THREE={MODAL\_CARD\_THREE} setActiveModal={setActiveModal} updateData={updateData} id='questions' go={go} />

                    <QuestionsList back={back} updateQuestion={updateQuestion} category={category} id='questions-list' go={go} />

                    <Instruction back={back} question={question} category={category} id='instruction' go={go} />

                    <CalendarPanel back={back} id='calendar' go={go} />

                </View>

            </AppRoot>

        </AdaptivityProvider>

    );

}

export default App;

Файл «AboutDirection.js»

import React, { Component } from 'react';

import Panel from '@vkontakte/vkui/dist/components/Panel/Panel';

import PanelHeader from '@vkontakte/vkui/dist/components/PanelHeader/PanelHeader';

import Div from '@vkontakte/vkui/dist/components/Div/Div';

import List from '@vkontakte/vkui/dist/components/List/List';

import { Cell } from '@vkontakte/vkui/dist/components/Cell/Cell';

import FixedLayout from '@vkontakte/vkui/dist/components/FixedLayout/FixedLayout';

import Link from '@vkontakte/vkui/dist/components/Link/Link';

import '../css/PickDirections\_v2.css';

import Title from '@vkontakte/vkui/dist/components/Typography/Title/Title';

import Text from '@vkontakte/vkui/dist/components/Typography/Text/Text';

import { PanelHeaderBack } from '@vkontakte/vkui';

import '../css/aboutDirection.css';

class AboutDirection extends Component {

    constructor(props) {

        super(props);

        this.state = {

            interestedDirection: ''

        }

    }

    componentDidMount = () => {

        this.setState({ interestedDirection: JSON.parse(localStorage.getItem('interestedDirection')) })

    }

    createMarkup = (text) => {

        return { \_\_html: text };

      }

    render() {

        return (

            <Panel id={this.props.id}>

                <PanelHeader left={<PanelHeaderBack onClick={this.props.back} />}>PolyApp</PanelHeader>

                <Title style={{ marginLeft: 16, marginRight: 16, marginTop: 16, padding: 16, background: 'var(--content\_tint\_background)', borderRadius: 7, fontSize: '1.2rem', textAlign: 'center' }}>

                    {this.state.interestedDirection['Название направления']}

                </Title>

                <Title level='2' weight="regular" style={{ marginLeft: 16, marginRight: 16, marginTop: 16, padding: '16 0 16 0', borderRadius: 7, textAlign: 'left', color: 'var(--text-primary)' }}>

                    о направлении:

        </Title>

                <List>

                    <Cell indicator={this.state.interestedDirection['Код направления']}>Код</Cell>

                    <Cell indicator={this.state.interestedDirection['Срок обучения']}>Срок обучения</Cell>

                    <Cell indicator={this.state.interestedDirection['Бюджетных мест']}>Бюджетных мест</Cell>

                    <Cell indicator={this.state.interestedDirection['Платных мест']}>Платных мест</Cell>

                </List>

                <Text weight="medium" style={{ marginBottom: 16, padding: 16, marginBlockEnd: 70}} dangerouslySetInnerHTML={this.createMarkup(this.state.interestedDirection['Описание'])}></Text>

                <FixedLayout filled vertical="bottom">

                <Div style={{ display: 'flex', flexDirection: 'row', alignItems: 'center', justifyContent: 'center!important' }}>

                    <Link style={{width: '100%', textAlign: 'center', color:"var(--button\_primary\_foreground)",

                    background:"var(--button\_primary\_background)", borderRadius:8, minHeight:44, display: "flex", justifyContent: "center", alignItems: "center", fontWeight:"500"}}

                    href={this.state.interestedDirection['Подробнее']} target="\_blank"> Подробнее</Link>

                    </Div>

                </FixedLayout>

            </Panel>

        )

    };

}

export default AboutDirection;

Файл «AboutStudent.js»

import React, { Component } from 'react';

import Panel from '@vkontakte/vkui/dist/components/Panel/Panel';

import PanelHeader from '@vkontakte/vkui/dist/components/PanelHeader/PanelHeader'

import { FormItem } from '@vkontakte/vkui/dist/components/FormItem/FormItem';

import Input from '@vkontakte/vkui/dist/components/Input/Input';

import { Button } from '@vkontakte/vkui';

import Select from '@vkontakte/vkui/dist/components/Select/Select';

import CustomSelectOption from '@vkontakte/vkui/dist/components/CustomSelectOption/CustomSelectOption';

import FixedLayout from '@vkontakte/vkui/dist/components/FixedLayout/FixedLayout';

import Div from '@vkontakte/vkui/dist/components/Div/Div';

import { PanelHeaderBack } from '@vkontakte/vkui';

import Radio from '@vkontakte/vkui/dist/components/Radio/Radio';

import $ from 'jquery';

import 'jquery-mask-plugin/dist/jquery.mask.min.js';

import '../css/AboutStudents.css'

const newDirections = require('../json/new\_directions.json');

class AboutStudent extends Component {

    constructor(props) {

        super(props);

        this.state = {

            level: '',

            year: '',

            specialty: '',

            group: '',

            dorm: '',

            dormnum: '',

            stud: '',

            prof: '',

            changed: false,

            groupValid: true,

            studValid: true,

            profValid: true,

        }

        this.handleInputChange = this.handleInputChange.bind(this);

    }

    componentDidMount = () => {

        localStorage.clear();

        $('input[name="group"]').mask('999-999');

        $('input[name="stud"]').mask('9990-9999');

        $('input[name="prof"]').mask('9999999999999999');

    }

    handleInputChange(event) {

        const target = event.target;

        const value = target.value;

        const name = target.name;

        this.setState({

            [name]: value,

        },

            () => { this.validateField(name, value) }

        );

    };

    validateField = (fieldName, value) => {

        let groupValid = this.state.groupValid;

        let profValid = this.state.profValid;

        let studValid = this.state.studValid;

        switch (fieldName) {

            case 'group':

                groupValid = (value.length == 7);

                break;

            case 'prof':

                profValid = (value.length == 16) || (value == '');

                break;

            case 'stud':

                studValid = (value.length == 9) || (value == '');

                break;

            default:

                break;

        }

        this.setState({

            groupValid: groupValid,

            profValid: profValid,

            studValid: studValid,

        }, this.validateForm);

    };

    validateForm = () => {

        this.setState({

            changed: this.state.groupValid &&

                this.state.profValid && this.state.studValid && (this.state.year !== '') && ((this.state.level === 'Бакалавриат' && this.state.year != 5)||(this.state.level === 'Магистратура' && this.state.year < 3)||this.state.level === 'Специалитет' )&&

                (this.state.level !== '')&&(this.state.specialty !== '') && (this.state.dorm !== '') && (((this.state.dormnum !== '') && (this.state.dorm == 'Yes')) || (this.state.dorm == 'No'))

        });

    };

    thirdhand(event){

        this.setState({specialty: ''});

        var inputs = document.getElementsByTagName('input');

        for (var i=0; i<inputs.length; i++)  {

        if (inputs[i].type == 'radio')   {

            inputs[i].checked = false;

        }

        }

        this.handleInputChange(event)

    }

    toStorage = () => {

        const {level, year, specialty, group, dorm, stud, prof, dormnum } = this.state;

        localStorage.setItem('level', level);

        localStorage.setItem('year', year);

        localStorage.setItem('specialty', specialty);

        localStorage.setItem('group', group);

        localStorage.setItem('dorm', dorm);

        localStorage.setItem('stud', stud);

        localStorage.setItem('prof', prof);

        localStorage.setItem('dormnum', dormnum);

    };

    render() {

        return (

            <Panel id={this.props.id}>

                <PanelHeader left={<PanelHeaderBack onClick={this.props.back} />}>PolyApp</PanelHeader>

                <FormItem top="Ступень">

                    <Select name="level" value={this.state.level}

                        onChange={(event)=>this.thirdhand(event)}

                        placeholder="Не выбран"

                        options={[{ value: 'Бакалавриат', label: 'Бакалавриат' }, { value: 'Специалитет', label: 'Специалитет' }, { value: 'Магистратура', label: 'Магистратура' }]}

                        renderOption={({ option, ...restProps }) => (

                            <CustomSelectOption {...restProps} />

                        )}

                    />

                </FormItem>

                {this.state.level === 'Бакалавриат'?

                    <FormItem top="Курс">

                        <Select name="year" value={this.state.year}

                            onChange={this.handleInputChange}

                            placeholder="Не выбран"

                            options={[{ value: 1, label: '1 курс' }, { value: 2, label: '2 курс' }, { value: 3, label: '3 курс' }, { value: 4, label: '4 курс' }]}

                            renderOption={({ option, ...restProps }) => (

                                <CustomSelectOption {...restProps} />

                            )}

                        />

                    </FormItem>

                :this.state.level === 'Специалитет'?

                    <FormItem top="Курс">

                        <Select name="year" value={this.state.year}

                            onChange={this.handleInputChange}

                            placeholder="Не выбран"

                            options={[{ value: 1, label: '1 курс' }, { value: 2, label: '2 курс' }, { value: 3, label: '3 курс' }, { value: 4, label: '4 курс' }, { value: 5, label: '5 курс' }]}

                            renderOption={({ option, ...restProps }) => (

                                <CustomSelectOption {...restProps} />

                            )}

                        />

                    </FormItem>

                :this.state.level === 'Магистратура'?

                    <FormItem top="Курс">

                        <Select name="year" value={this.state.year}

                            onChange={this.handleInputChange}

                            placeholder="Не выбран"

                            options={[{ value: 1, label: '1 курс' }, { value: 2, label: '2 курс' }]}

                            renderOption={({ option, ...restProps }) => (

                                <CustomSelectOption {...restProps} />

                            )}

                        />

                    </FormItem>:null

                }

                {this.state.level !== ''?

                    newDirections.Очная[this.state.level].map((directions, index)=>{

                        return(

                            <FormItem key={index}>

                            {directions[Object.keys(directions)].map((direction, idx) =>{

                                return(

                                    <Radio key={idx} onChange={this.handleInputChange} name="specialty" value={direction['Название направления']}>{direction['Название направления']}</Radio>

                                )

                            })}

                        </FormItem>

                        )

                    })

                :null

                }

                <FormItem top="Группа">

                    <Input type="text" inputMode={'numeric'} name="group" className={!this.state.groupValid && this.state.group != '' ? 'red' : null}

                        autoСomplete="off"

                        value={this.state.group}

                        onChange={this.handleInputChange}

                        placeholder="000000" />

                </FormItem>

                <FormItem top="Живешь в общежитии">

                    <Select value={this.state.dorm}

                        onChange={this.handleInputChange}

                        name='dorm'

                        placeholder="Не выбрано"

                        options={[{ value: 'Yes', label: 'Да' }, { value: 'No', label: 'Нет' }]}

                        renderOption={({ option, ...restProps }) => (

                            <CustomSelectOption {...restProps} />

                        )}

                    />

                </FormItem>

                {(this.state.dorm != '') && (this.state.dorm != 'No') ?

                    <FormItem top="Номер общежития">

                        <Select value={this.state.dormnum}

                            onChange={this.handleInputChange}

                            name='dormnum'

                            placeholder="Не выбрано"

                            options={[{ value: '1', label: 'Общежитие номер 1' }, { value: '2', label: 'Общежитие номер 2' }, { value: '3', label: 'Общежитие номер 3' }, { value: '4', label: 'Общежитие номер 4' }, { value: '5', label: 'Общежитие номер 5' }, { value: '6', label: 'Общежитие номер 6' }, { value: '7', label: 'Общежитие номер 7' }, { value: '8', label: 'Общежитие номер 8' }, { value: '9', label: 'Общежитие номер 9' }, { value: '10', label: 'Общежитие номер 10' }]}

                            renderOption={({ option, ...restProps }) => (

                                <CustomSelectOption {...restProps} />

                            )}

                        />

                    </FormItem>

                    : null

                }

                <FormItem top="Номер студенческого(необязательно)">

                    <Input type="text" name="stud" inputMode={'numeric'} className={!this.state.studValid ? 'red' : null}

                        autoСomplete="off"

                        value={this.state.stud}

                        onChange={this.handleInputChange}

                        placeholder="0000-0000" />

                </FormItem>

                <FormItem top="Номер профбилета(необязательно)" style={{ marginBlockEnd: 70, writingMode: 'horizontal-tb' }}>

                    <Input type="text" inputMode={'numeric'} name="prof" className={!this.state.profValid ? 'red' : null}

                        autoСomplete="off"

                        value={this.state.prof}

                        onChange={this.handleInputChange}

                        placeholder="0000000000000000" />

                </FormItem>

                <FixedLayout filled vertical="bottom">

                    <Div>

                        <Button type="submit" stretched size="l" mode="primary" disabled={!this.state.changed}

                            onClick={this.toStorage(), this.props.go} data-to="home">Продолжить</Button>

                    </Div>

                </FixedLayout>

            </Panel>

        )

    }

}

export default AboutStudent;

Файл «Acquaintance.js»

import React, { useEffect, useRef, useState } from 'react';

import PropTypes from 'prop-types';

import Panel from '@vkontakte/vkui/dist/components/Panel/Panel';

import PanelHeader from '@vkontakte/vkui/dist/components/PanelHeader/PanelHeader';

import Button from '@vkontakte/vkui/dist/components/Button/Button';

import Group from '@vkontakte/vkui/dist/components/Group/Group';

import Div from '@vkontakte/vkui/dist/components/Div/Div';

import Caption from '@vkontakte/vkui/dist/components/Typography/Caption/Caption';

import FormLayout from '@vkontakte/vkui/dist/components/FormLayout/FormLayout';

import Radio from '@vkontakte/vkui/dist/components/Radio/Radio';

import { FormItem } from '@vkontakte/vkui/dist/components/FormItem/FormItem';

import FixedLayout from '@vkontakte/vkui/dist/components/FixedLayout/FixedLayout';

import { Cell } from '@vkontakte/vkui/dist/components/Cell/Cell';

import Subhead from '@vkontakte/vkui/dist/components/Typography/Subhead/Subhead';

import Header from '@vkontakte/vkui/dist/components/Header/Header';

import Avatar from '@vkontakte/vkui/dist/components/Avatar/Avatar';

import { Icon20QuestionOutline } from '@vkontakte/icons';

import '../css/Radio.css';

import { scrollToBottom } from 'react-scroll/modules/mixins/animate-scroll';

import { Title } from '@vkontakte/vkui';

var Scroll = require('react-scroll');

var scroll = Scroll.animateScroll;

const Acquaintance = ({ id, go, fetchedUser, setActiveModal, MODAL\_CARD\_ONE, MODAL\_CARD\_TWO }) => {

    const [want, setWant] = useState(false)

    const [wantStudyForm, setForm] = useState('Очная')

    const [wantStudyLevel, setLevel] = useState('Бакалавриат')

    useEffect(() => {

        localStorage.setItem('selectedGroups', []);

    });

    const handleNextClick = (event) => {

        localStorage.setItem('wantStudyForm', wantStudyForm);

        localStorage.setItem('wantStudyLevel', wantStudyLevel);

        go(event)

    };

    const scrollToSection = () => {

        setWant(true)

        scroll.scrollToBottom();

    };

    return (

        <Panel id={id}>

            <PanelHeader>PolyApp</PanelHeader>

            {fetchedUser &&

                <Group header={<Header mode="primary">Привет!</Header>}>

                    <Div className="homepage-subhead">

                        <Subhead weight="semibold">Это приложение для абитуриентов и студентов факультета информационных технологий Московского политеха.</Subhead>

                    </Div>

                    <Cell

                        before={fetchedUser.photo\_200 ? <Avatar src={fetchedUser.photo\_200} /> : null}

                        description={fetchedUser.city && fetchedUser.city.title ? fetchedUser.city.title : ''}

                    >

                        {`${fetchedUser.first\_name} ${fetchedUser.last\_name}`}

                    </Cell>

                </Group>}

            <Group>

                <Div>

                    <FormLayout>

                        <FormItem top="Расскажи про себя">

                            <Radio name="radio" onClick={() => setWant(false)} value="already" defaultChecked>Я уже студент</Radio>

                            <Radio name="radio" onClick={() => scrollToSection()} value="newStudent">Я хочу поступить</Radio>

                        </FormItem>

                    </FormLayout>

                </Div>

                {want ?

                    <Div style={{ marginBlockEnd: 50 }}>

                        <Caption className="captionCaps" level="1" weight="semibold" caps >

                            Круто, хороший выбор!

                        </Caption>

                        <FormLayout>

                            <Header mode="primary" aside={<Icon20QuestionOutline onClick={() => setActiveModal(MODAL\_CARD\_ONE)} />} style={{ marginBottom: 10 }}>Форма обучения</Header>

                            <FormItem><Radio name="radio" value="Очная" onClick={() => setForm('Очная')} defaultChecked>Очная</Radio>

                                <Radio name="radio" value="Заочная" onClick={() => setForm('Заочная')}>Заочная</Radio></FormItem>

                        </FormLayout>

                        {wantStudyForm == 'Очная' ?

                            <FormLayout>

                                <Header mode="primary" style={{ marginBottom: 10 }} aside={<Icon20QuestionOutline onClick={() => setActiveModal(MODAL\_CARD\_TWO)} />}>Ступень образования</Header>

                                <FormItem><Radio name="radio" value="Бакалавриат" onClick={() => setLevel('Бакалавриат')} defaultChecked>Бакалавриат</Radio>

                                    <Radio name="radio" value="Специалитет" onClick={() => setLevel('Специалитет')} >Специалитет</Radio>

                                    <Radio name="radio" value="Магистратура" onClick={() => setLevel('Магистратура')} >Магистратура</Radio></FormItem>

                            </FormLayout>

                            :

                            <FormLayout>

                                <Header mode="primary" style={{ marginBottom: 10 }} aside={<Icon20QuestionOutline onClick={() => setActiveModal(MODAL\_CARD\_TWO)} />}>Ступень образования</Header>

                                    <FormItem><Radio name="radio" value="Бакалавриат" onClick={() => setLevel('Бакалавриат')} defaultChecked>Бакалавриат</Radio></FormItem>

                            </FormLayout>

                        }

                    </Div>

                    : null

                }

                <FixedLayout filled vertical="bottom">

                    <Div>

                        <Button stretched size="l" mode="primary" onClick={(event) => handleNextClick(event)} data-to={want ? 'pick-directions' : 'student-form-filling'} >

                            Далее

                        </Button>

                    </Div>

                </FixedLayout>

            </Group>

        </Panel>

    );

}

Acquaintance.propTypes = {

    id: PropTypes.string.isRequired,

    go: PropTypes.func.isRequired,

    fetchedUser: PropTypes.shape({

        photo\_200: PropTypes.string,

        first\_name: PropTypes.string,

        last\_name: PropTypes.string,

        city: PropTypes.shape({

            title: PropTypes.string,

        }),

    }),

};

export default Acquaintance;

Файл «ChoosedDirectionsInfo.js»

import React, { Component } from 'react';

import Panel from '@vkontakte/vkui/dist/components/Panel/Panel';

import PanelHeader from '@vkontakte/vkui/dist/components/PanelHeader/PanelHeader';

import { PanelHeaderBack } from '@vkontakte/vkui';

import Div from '@vkontakte/vkui/dist/components/Div/Div';

import FixedLayout from '@vkontakte/vkui/dist/components/FixedLayout/FixedLayout';

import Button from '@vkontakte/vkui/dist/components/Button/Button';

import '../css/PickDirections\_v2.css';

import Title from '@vkontakte/vkui/dist/components/Typography/Title/Title';

import Text from '@vkontakte/vkui/dist/components/Typography/Text/Text';

let newDirections = require('../json/new\_directions.json');

class ChoosedDirectionsInfo extends Component {

    constructor(props) {

        super(props);

        this.state = {

            messageDirections: `\n`,

            messageDirectionsArray: [],

            requiredPoint: {

                'Математика': '39',

                'Русский язык': '40',

                "Информатика и ИКТ/Физика": "44/39"

            }

        }

    }

    componentDidMount = () => {

        this.state.messageDirectionsArray = localStorage.getItem('choosedGroups').split(',');

    }

    render() {

        return (

            <Panel id={this.props.id}>

                <PanelHeader left={<PanelHeaderBack onClick={this.props.back} />}>PolyApp</PanelHeader>

                <Title className="choosedDirectionInfo\_title">

                    Отлично! Ты выбрал:

                            {this.componentDidMount()}

                            <ul>

                                {this.state.messageDirectionsArray.map((unit, index) => {

                                    return (

                                        <li style={{ fontSize: '1.2rem' }} key={index}>{unit}</li>

                                    )

                                }

                                )}

                            </ul>

                </Title>

                <Title className="choosedDirectionInfo\_title">

                Чтобы участвовать в конкурсе на поступление тебе нужно набрать минимум по профильным предметам и пройти конкурс:

                    <ul>    {localStorage.getItem('wantStudyLevel') == 'magistracy' ?

                        <Text>- Междисциплинарный экзамен — 40</Text>

                        : Object.keys(this.state.requiredPoint).map((key, index) => {

                            return (

                                <li  key={index}> {key} {this.state.requiredPoint[key]}</li>

                            )

                        }

                        )}

                    </ul>

                </Title>

                <Title className="choosedDirectionInfo\_title">

                    Как подать документы?

                    <ul>

                        <li>Через <a className='link-dir' href='https://lk.mospolytech.ru/user/sign-in/login' target='blank'>личный кабинет абитуриента Московского Политеха</a></li>

                        <li>Через <a className='link-dir' href='https://www.gosuslugi.ru/10077/1' target='blank'>портал Госуслуг</a></li>

                        <li>Лично в университете</li>

                    </ul>

                </Title>

                <Title className="choosedDirectionInfo\_title">

                    Подготовить документы:

                    <ul>

                        <li>Копия паспорта</li>

                        <li>Аттестат или диплом СПО</li>

                        <li>Копия СНИЛС</li>

                        <li>Заявление на поступление</li>

                        <li>Документы подтверждающие льготы (при наличии)</li>

                        <li>2 матовые фотографии 3X4 (для поступающих по ВИ)</li>

                        <li>Документы подтвержающие участие в олимпиадах (при наличии)</li>

                    </ul>

                </Title>

                <Title className="choosedDirectionInfo\_title">

                    Приём документов начинается 20 июня. Ждём тебя!

                </Title>

                <FixedLayout className="choosedDirectionInfo\_\_fixed" filled vertical="bottom">

                    <Div>

                        <Button stretched size="l" mode="primary" onClick={this.props.go} data-to="dorms">Общежития</Button>

                    </Div>

                </FixedLayout>

            </Panel>

        )

    };

}

export default ChoosedDirectionsInfo;

Файл «Calendar.js»

import React, { useEffect, useState } from 'react';

import bridge from '@vkontakte/vk-bridge';

import Panel  from '@vkontakte/vkui/dist/components/Panel/Panel';

import PanelHeader from '@vkontakte/vkui/dist/components/PanelHeader/PanelHeader';

import Button from '@vkontakte/vkui/dist/components/Button/Button';

import Group from '@vkontakte/vkui/dist/components/Group/Group';

import {Cell} from '@vkontakte/vkui/dist/components/Cell/Cell';

import Div from '@vkontakte/vkui/dist/components/Div/Div';

import Avatar from '@vkontakte/vkui/dist/components/Avatar/Avatar';

import { FormItem, Input } from '@vkontakte/vkui';

import { Tabbar, TabbarItem } from '@vkontakte/vkui';

import { Icon28UserCircleOutline } from '@vkontakte/icons';

import { Icon28CalendarOutline } from '@vkontakte/icons';

import { Icon28InfoCircleOutline } from '@vkontakte/icons';

import FixedLayout from '@vkontakte/vkui/dist/components/FixedLayout/FixedLayout';

import DatePicker from '@vkontakte/vkui/dist/components/DatePicker/DatePicker';

import DatePicker\_ from 'react-date-picker';

import Calendar from 'react-calendar';

import '../css/Calendar.css';

const CalendarPanel = ({ id, go, back}) => {

    const [notifies, setNotifies] = useState({});

    const [value, setValue] = useState('');

    const [text, setText] = useState([]);

    const [date, setDate] = useState();

    const [allowedNotifies, setAllow] = useState(false);

    const copyText = () => {

        bridge.send('VKWebAppCopyText', {text: value})

    }

    const saveText = async (text) => {

        setValue(text);

        await bridge.send('VKWebAppStorageSet', {

            key: 'someKey',

            value: JSON.stringify(text)

        });

    }

    async function deleteAllNotifies() {

        try {

            await bridge.send('VKWebAppStorageSet', {

                key: 'notifies',

                value: ''

            });

            setNotifies({});

        }

        catch (error) {

            console.error(error);

        }

        console.log('notifies removed');

    }

    async function getAllNotifies() {

        if (Object.keys(notifies).length) console.log(notifies);

        else {

            try {

            const storageData = await bridge.send('VKWebAppStorageGet', {keys: ['notifies']});

            const value = JSON.parse(storageData.keys[0].value);

            setNotifies(value);

            console.log(value);

            }

            catch (error) {

                console.error(error);

            }

        }

    }

    async function showNotifies(date) {

        const key = date.getDate() + '/' + (date.getMonth()+1) + '/' + date.getFullYear();

        if (Object.keys(notifies).length) {

            if (!notifies[key]) {

                console.log('Нет напоминаний');

                setText('Нет напоминаний');

                return;

            }

            let s = key + '\n' + Object.keys((notifies[key])).join('\n');

            console.log('This is s' + s);

            setText(s);

        }

        else {

            try {

                const storageData = await bridge.send('VKWebAppStorageGet', {keys: ['notifies']});

                const notifies = JSON.parse(storageData.keys[0].value);

                if (!notifies || !Object.keys(notifies).length) {

                    console.log("No notifies today");

                    return;

                }

                setNotifies(notifies);

                if (!notifies[key]) {

                    console.log('Нет напоминаний');

                    setText('Нет напоминаний');

                    return;

                }

                let s = key + '\n' + Object.keys((notifies[key])).join('\n');

                console.log(s);

                setText(s);

            } catch (error) {

                console.error(error);

                console.log('no any notify');

            }

        }

    }

    async function addNotifies(choosedDate, message) {

        if (!choosedDate || !choosedDate.day || !choosedDate.month || !choosedDate.year) {

            console.error('Введите дату');

            setText('Введите дату');

            return;

        }

        console.log('raw date: ' + Object.keys(choosedDate));

        const key = choosedDate.day + '/' + choosedDate.month + '/' + choosedDate.year;

        console.log('parsed date: ' + key);

        console.log('mesage: ' + message);

        setText([message, ...text])

        console.log('notifies: ' + notifies);

        console.log('notifies\_keys: ' + Object.keys(notifies));

        if (!notifies[key]) notifies[key] = {};

        notifies[key][message] = "12:00:00";

        try{

            await bridge.send('VKWebAppStorageSet', {

                key: 'notifies',

                value: JSON.stringify(notifies)

            });

            setNotifies({});

            setNotifies(notifies);

        }

        catch (error) {

            console.error('error:' + error);

        }

        console.log('notify added');

    }

    function highlightDates(date) {

        const parsedDate = date.getDate() + '/' + (date.getMonth()+1) + '/' + date.getFullYear();

        if (Object.keys(notifies).find(notifyDate => parsedDate == notifyDate))

            return 'highlight';

    }

    useEffect(() => {

        async function getText() {

            try {

            const storageData = await bridge.send('VKWebAppStorageGet', {keys: ['someKey']});

            setValue(JSON.parse(storageData.keys[0].value));

            } catch(error) {

                console.error('error:', error);

            }

    }

    getText();

    getAllNotifies();

    }, []);

    return (

        <Panel id={id}>

            <PanelHeader>PolyApp</PanelHeader>

            <Group>

                <Div>

                    <Button size='m' mode='primary' onClick={getAllNotifies}>

                        Уведомления

                    </Button>

                    <Button size='m' mode='primary' onClick={() => addNotifies(date, document.getElementById('input').value)}>

                        Добавить уведомление

                    </Button>

                    <Button size='m' mode='primary' onClick={() => deleteAllNotifies()}>

                        Удалить уведомления

                    </Button>

                    <FormItem >

                    <Input id='input' type='text' placeholder='Hello world' onChange={(e) => saveText(e.target.value)} defaultValue={value}/>

                    </FormItem>

                </Div>

                <DatePicker

                min={{day: 11, month: 6, year: 2021}}

                max={{day: 1, month: 1, year: 2024}}

                popupDirection='bottom'

                onDateChange={(value) => {console.log(value); setDate(value)}}

                dayPlaceholder="ДД"

                monthPlaceholder="ММММ"

                yearPlaceholder="ГГГГ"/>

                <Div style={{whiteSpace: 'pre-line'}}>

                    {text}

                </Div>

            </Group>

                <FixedLayout filled vertical="bottom">

                <Tabbar className='tabbar-padding'>

                    <TabbarItem text="Вопросы" onClick={go} data-to="questions">

                        <Icon28InfoCircleOutline/>

                    </TabbarItem>

                    <TabbarItem text="Календарь" selected>

                        <Icon28CalendarOutline />

                    </TabbarItem>

                    <TabbarItem text="Профиль" onClick={go} data-to="home">

                        <Icon28UserCircleOutline/>

                    </TabbarItem>

                </Tabbar>

            </FixedLayout>

        </Panel>

    );

}

export default CalendarPanel;

Файл «Degree.js»

import React from 'react';

import PropTypes from 'prop-types';

import Panel from '@vkontakte/vkui/dist/components/Panel/Panel';

import PanelHeader from '@vkontakte/vkui/dist/components/PanelHeader/PanelHeader';

import Button from '@vkontakte/vkui/dist/components/Button/Button';

import Group from '@vkontakte/vkui/dist/components/Group/Group';

import Div from '@vkontakte/vkui/dist/components/Div/Div';

import Caption from '@vkontakte/vkui/dist/components/Typography/Caption/Caption';

import FormLayout from '@vkontakte/vkui/dist/components/FormLayout/FormLayout';

import Radio from '@vkontakte/vkui/dist/components/Radio/Radio';

import {FormItem} from '@vkontakte/vkui/dist/components/FormItem/FormItem';

import FixedLayout from '@vkontakte/vkui/dist/components/FixedLayout/FixedLayout';

import '../css/Radio.css';

const Degree = ({ id, go }) => {

    return(

    <Panel id={id}>

        <PanelHeader>PolyApp</PanelHeader>

        <Group>

            <Div>

                <Caption className="captionCaps" level="1" weight="semibold" caps >

                    Отлично! А теперь выбери ступень высшего образования.

                </Caption>

                <FormLayout>

                    <FormItem top="Форма обуение">

                        <Radio name="radio" value="bachelor" defaultChecked>Бакалавриат</Radio>

                        <Radio name="radio" value="magistracy">Магистратура</Radio>

                    </FormItem>

                </FormLayout>

            </Div>

            <FixedLayout filled vertical="bottom">

                <Div>

                    <Button stretched size="l" mode="primary" onClick={go} data-to="pick-directions">

                            Далее

                    </Button>

                </Div>

            </FixedLayout>

        </Group>

    </Panel>

)

};

Degree.propTypes = {

    id: PropTypes.string.isRequired,

    go: PropTypes.func.isRequired,

};

export default Degree;

Файл «DormPage.js»

import React, { Component } from 'react';

import Panel from '@vkontakte/vkui/dist/components/Panel/Panel';

import PanelHeader from '@vkontakte/vkui/dist/components/PanelHeader/PanelHeader';

import Div from '@vkontakte/vkui/dist/components/Div/Div';

import FixedLayout from '@vkontakte/vkui/dist/components/FixedLayout/FixedLayout';

import Button from '@vkontakte/vkui/dist/components/Button/Button';

import '../css/PickDirections\_v2.css';

import Title from '@vkontakte/vkui/dist/components/Typography/Title/Title';

import Text from '@vkontakte/vkui/dist/components/Typography/Text/Text';

import { Gallery, Group, PanelHeaderBack } from '@vkontakte/vkui';

let dorms = require('../json/dorms.json');

import '../css/dorm.css';

class DormPage extends Component {

    constructor(props) {

        super(props);

        this.state = {

        }

    }

    createMarkup = (text) => {

      return {\_\_html: text};

    }

    goNextPageHandler = () => {

      this.props.dorm ===  dorms['Общежития'].length -1 ? this.props.setdorm(0) :this.props.setdorm(this.props.dorm +1 );

    }

    goBackPageHandler = () =>{

      this.props.dorm === 0 ?this.props.setdorm(dorms['Общежития'].length -1 ) : this.props.setdorm(this.props.dorm -1 );

    }

    render() {

        return (

            <Panel id={this.props.id}>

                <PanelHeader left={<PanelHeaderBack onClick={this.props.back} />}>PolyApp</PanelHeader>

                <Title level="1" weight="regular" className="dorm\_\_title">Общежитие {dorms.Общежития[`${this.props.dorm}`].Номер} </Title>

                <Group>

                <Gallery

                  slideWidth="90%"

                  style={{ height: 300 }}

                  bullets="dark"

                  showArrows={true}

                  isDraggable={true}

                >

                  {

                dorms.Общежития[this.props.dorm].Фотографии.map((photo,index)=>{

                  console.log(photo)

                  return (

                    <div style={{backgroundImage: `url(${photo})`, backgroundSize: 'cover'}} key={index}/>

                  )

                })

              }

                </Gallery>

              </Group>

            <Div style={{margin:'0 16px 8px 16px', lineHeight:'1.4rem', marginBlockEnd: 70}}>

                <Text dangerouslySetInnerHTML={this.createMarkup(dorms.Общежития[this.props.dorm].Описание)} />

               </Div>

               <FixedLayout filled vertical="bottom">

                <Div style={{display: "flex", justifyContent: "space-between"}}>

                  <Button  style={{width: "40%"}} size="l"  onClick={this.goBackPageHandler} data-to="choosed-directions-info">Предыдущее</Button>

                  <Button  style={{width: "40%"}} size="l"  onClick={this.goNextPageHandler} data-to="choosed-directions-info">Следующее</Button>

                </Div>

              </FixedLayout>

            </Panel>

        )

    };

}

export default DormPage;

Файл «Dorms.js»

import React, { Component } from 'react';

import Panel from '@vkontakte/vkui/dist/components/Panel/Panel';

import PanelHeader from '@vkontakte/vkui/dist/components/PanelHeader/PanelHeader';

import { PanelHeaderBack } from '@vkontakte/vkui';

import Group from '@vkontakte/vkui/dist/components/Group/Group';

import Title from '@vkontakte/vkui/dist/components/Typography/Title/Title';

import Text from '@vkontakte/vkui/dist/components/Typography/Text/Text';

import Card from '@vkontakte/vkui/dist/components/Card/Card';

import CardGrid from '@vkontakte/vkui/dist/components/CardGrid/CardGrid';

import { Icon16New } from '@vkontakte/icons';

const dormsInfo = require('../json/dorms.json');

import '../css/dorms.css';

class Dorms extends Component {

  constructor(props) {

      super(props);

      this.state = {

          interestedDorm: ''

      }

    }

    handeDormClick = (id) =>(event)=> {

      this.props.setdorm(id);

      this.props.go(event)

    }

    createMarkup = (text) => {

      return {\_\_html: text};

    }

  render(){

    return(

      <Panel id={this.props.id}>

        <PanelHeader left={<PanelHeaderBack onClick={this.props.back} />}>PolyApp</PanelHeader>

        <Group>

          <Title level="1" weight="regular" className="dorms\_\_title">Общежития</Title>

          <Text className="dorms\_\_text">

            В Московском Политехе студентам предоставляется место в общежитии, если они проживают за пределами трассы А107

          </Text>

          <Text className="dorms\_\_text">

            Выбор общежития предоставляется в приоритетном порядке:

          </Text>

          <ul className="dorms\_\_priority">

            <li> Студенты со средним баллом 85+ </li>

            <li> Студенты поступившие на бюджет </li>

            <li> Студенты платной формы обучения со средним баллом 85+ </li>

            <li> Студенты платной формы обучения</li>

          </ul>

        </Group>

        <Group style={{marginTop:8}}>

        <CardGrid size="l">

              {

                dormsInfo.Общежития.map((dorm,index)=>{

                  return (

                    <Card  key={index} className="dorms\_\_card" onClick={this.handeDormClick(index)} data-to='dorm-page'>

                    <div className="dorms\_\_card\_\_photo" style={{background:"url(" + dorm.Фотографии[0]+ (')')}}>

                    </div>

                    <div className="dorms\_\_card\_\_content">

                      <Text className="dorms\_\_card\_\_content\_\_number">Общежитие {dorm.Номер}</Text>

                      <Text className="dorms\_\_card\_\_content\_\_address"> {dorm['Адрес']} </Text>

                      <div className="dorms\_\_card\_\_content\_\_metro">

                      <Icon16New className="dorms\_\_card\_\_content\_\_metro\_\_color" fill={dorm.Цвет}/>

                      <Text className="dorms\_\_card\_\_content\_\_metro\_\_text" style={{textAlign:'left', opacity:.5, paddingTop: '0!important'}} >{dorm['Метро']}</Text>

                      </div>

                    </div>

                  </Card>

                  )

                })

              }

            </CardGrid>

        </Group>

      </Panel>

    )

  }

}

export default Dorms;

Файл «EditStudent.js»

import React, { Component } from 'react';

import Panel from '@vkontakte/vkui/dist/components/Panel/Panel';

import PanelHeader from '@vkontakte/vkui/dist/components/PanelHeader/PanelHeader';

import { FormItem } from '@vkontakte/vkui/dist/components/FormItem/FormItem';

import Input from '@vkontakte/vkui/dist/components/Input/Input';

import { Button } from '@vkontakte/vkui';

import Select from '@vkontakte/vkui/dist/components/Select/Select';

import CustomSelectOption from '@vkontakte/vkui/dist/components/CustomSelectOption/CustomSelectOption';

import FixedLayout from '@vkontakte/vkui/dist/components/FixedLayout/FixedLayout';

import Div from '@vkontakte/vkui/dist/components/Div/Div';

import { PanelHeaderBack } from '@vkontakte/vkui';

import $ from 'jquery';

import 'jquery-mask-plugin/dist/jquery.mask.min.js';

import '../css/AboutStudents.css'

class EditStudent extends Component {

    constructor(props) {

        super(props);

        this.state = {

            year: '',

            specialty: '',

            group: '',

            dorm: '',

            dormnum: '',

            stud: '',

            prof: '',

            changed: false,

            groupValid: true,

            studValid: true,

            profValid: true,

        }

        this.handleInputChange = this.handleInputChange.bind(this);

    }

    componentDidMount = () => {

        const group = localStorage.getItem('group');

        const stud = localStorage.getItem('stud');

        const prof = localStorage.getItem('prof');

        const dorm = localStorage.getItem('dorm');

        const year = localStorage.getItem('year');

        const specialty = localStorage.getItem('specialty');

        const dormnum = localStorage.getItem('dormnum');

        this.setState({ stud, group, year, dorm, prof, specialty, dormnum });

        $('input[name="group"]').mask('999-999');

        $('input[name="stud"]').mask('9990-9999');

        $('input[name="prof"]').mask('9999999999999999');

    }

    handleInputChange(event) {

        const target = event.target;

        const value = target.value;

        const name = target.name;

        this.setState({

            [name]: value,

        },

            () => { this.validateField(name, value) }

        );

    };

    validateField = (fieldName, value) => {

        let groupValid = this.state.groupValid;

        let profValid = this.state.profValid;

        let studValid = this.state.studValid;

        switch (fieldName) {

            case 'group':

                groupValid = (value.length == 7);

                break;

            case 'prof':

                profValid = (value.length == 16) || (value == '');

                break;

            case 'stud':

                studValid = (value.length == 9) || (value == '');

                break;

            default:

                break;

        }

        this.setState({

            groupValid: groupValid,

            profValid: profValid,

            studValid: studValid,

        }, this.validateForm);

    };

    validateForm = () => {

        this.setState({

            changed: this.state.groupValid &&

                this.state.profValid && this.state.studValid && (this.state.year !== '') &&

                (this.state.specialty !== '') && (this.state.dorm !== '') && (((this.state.dormnum !== '') && (this.state.dorm == 'Yes')) || (this.state.dorm == 'No'))

        });

    };

    toStorage = (event) => {

        const { year, specialty, group, dorm, stud, prof, dormnum } = this.state;

        localStorage.setItem('year', year);

        localStorage.setItem('specialty', specialty);

        localStorage.setItem('group', group);

        localStorage.setItem('dorm', dorm);

        localStorage.setItem('stud', stud);

        localStorage.setItem('prof', prof);

        localStorage.setItem('dormnum', dormnum);

        this.props.setActivePanel(event.currentTarget.dataset.to)

    };

    render() {

        return (

            <Panel id={this.props.id}>

                <PanelHeader left={<PanelHeaderBack onClick={this.props.back} />}>PolyApp</PanelHeader>

                <FormItem top="Курс">

                    <Select name="year" value={this.state.year}

                        onChange={this.handleInputChange}

                        placeholder="Не выбран"

                        options={[{ value: 1, label: '1 курс' }, { value: 2, label: '2 курс' }, { value: 3, label: '3 курс' }, { value: 4, label: '4 курс' }]}

                        renderOption={({ option, ...restProps }) => (

                            <CustomSelectOption {...restProps} />

                        )}

                    />

                </FormItem>

                <FormItem top="Направление">

                    <Select name="specialty" value={this.state.specialty}

                        onChange={this.handleInputChange}

                        placeholder="Не выбрано"

                        options={[{ value: 'Информационная безопасность автоматизированных систем', label: 'Информационная безопасность автоматизированных систем' }, { value: 'Прикладная математика и информатика (Большие и открытые данные)', label: 'Прикладная математика и информатика (Большие и открытые данные)' }, { value: 'Веб-технологии', label: 'Веб-технологии' }, { value: 'Интеграция и программирование в САПР', label: 'Интеграция и программирование в САПР' }, { value: 'Программное обеспечение информационных систем', label: 'Программное обеспечение информационных систем' }, { value: 'Киберфизические системы', label: 'Киберфизические системы' }, { value: 'Большие и открытые данные', label: 'Большие и открытые данные' }, { value: 'Корпоративные информационные системы', label: 'Корпоративные информационные системы' }, { value: 'Информационная безопасность', label: 'Информационная безопасность' }]}

                        renderOption={({ option, ...restProps }) => (

                            <CustomSelectOption {...restProps} />

                        )}

                    />

                </FormItem>

                <FormItem top="Группа">

                    <Input type="text" inputMode={'numeric'} name="group" className={!this.state.groupValid && this.state.group != '' ? 'red' : null}

                        autocomplete="off"

                        value={this.state.group}

                        onChange={this.handleInputChange}

                        placeholder="000000" />

                </FormItem>

                <FormItem top="Живешь в общежитии">

                    <Select value={this.state.dorm}

                        onChange={this.handleInputChange}

                        name='dorm'

                        placeholder="Не выбрано"

                        options={[{ value: 'Yes', label: 'Да' }, { value: 'No', label: 'Нет' }]}

                        renderOption={({ option, ...restProps }) => (

                            <CustomSelectOption {...restProps} />

                        )}

                    />

                </FormItem>

                {(this.state.dorm != '') && (this.state.dorm != 'No') ?

                    <FormItem top="Номер общежития">

                        <Select value={this.state.dormnum}

                            onChange={this.handleInputChange}

                            name='dormnum'

                            placeholder="Не выбрано"

                            options={[{ value: '1', label: 'Общежитие номер 1' }, { value: '2', label: 'Общежитие номер 2' }, { value: '3', label: 'Общежитие номер 3' }, { value: '4', label: 'Общежитие номер 4' }, { value: '5', label: 'Общежитие номер 5' }, { value: '6', label: 'Общежитие номер 6' }, { value: '7', label: 'Общежитие номер 7' }, { value: '8', label: 'Общежитие номер 8' }, { value: '9', label: 'Общежитие номер 9' }, { value: '10', label: 'Общежитие номер 10' }]}

                            renderOption={({ option, ...restProps }) => (

                                <CustomSelectOption {...restProps} />

                            )}

                        />

                    </FormItem>

                    : null

                }

                <FormItem top="Номер студенческого(необязательно)">

                    <Input type="text" name="stud" inputMode={'numeric'} className={!this.state.studValid ? 'red' : null}

                        autocomplete="off"

                        value={this.state.stud}

                        onChange={this.handleInputChange}

                        placeholder="0000-0000" />

                </FormItem>

                <FormItem top="Номер профбилета(необязательно)" style={{ marginBlockEnd: 70, writingMode: 'horizontal-tb' }}>

                    <Input type="text" inputMode={'numeric'} name="prof" className={!this.state.profValid ? 'red' : null}

                        autocomplete="off"

                        value={this.state.prof}

                        onChange={this.handleInputChange}

                        placeholder="0000000000000000" />

                </FormItem>

                <FixedLayout filled vertical="bottom">

                    <Div>

                        <Button type="submit" stretched size="l" mode="primary" disabled={!this.state.changed}

                            onClick={this.toStorage} data-to="home">Продолжить</Button>

                    </Div>

                </FixedLayout>

            </Panel>

        )

    }

}

export default EditStudent;

Файл «HomePage.js»

import React, { Component } from 'react';

import Panel from '@vkontakte/vkui/dist/components/Panel/Panel';

import PanelHeader from '@vkontakte/vkui/dist/components/PanelHeader/PanelHeader';

import FixedLayout from '@vkontakte/vkui/dist/components/FixedLayout/FixedLayout';

import Avatar from '@vkontakte/vkui/dist/components/Avatar/Avatar';

import Div from '@vkontakte/vkui/dist/components/Div/Div';

import Header from '@vkontakte/vkui/dist/components/Header/Header';

import Group from '@vkontakte/vkui/dist/components/Group/Group';

import Gradient from '@vkontakte/vkui/dist/components/Gradient/Gradient';

import Text from '@vkontakte/vkui/dist/components/Typography/Text/Text';

import Title from '@vkontakte/vkui/dist/components/Typography/Title/Title';

import { Button, Tabbar, TabbarItem } from '@vkontakte/vkui';

import { Icon28UserCircleOutline } from '@vkontakte/icons';

import { Icon28CalendarOutline } from '@vkontakte/icons';

import { Icon28InfoCircleOutline } from '@vkontakte/icons';

import { Icon20Write } from '@vkontakte/icons';

import { Icon16Clear } from '@vkontakte/icons';

import '../css/Home.css'

class HomePage extends Component {

  constructor(props) {

    super(props);

    this.state = {

      group: '',

      level: '',

      stud: '',

      prof: '',

      dorm: '',

      year: '',

      specialty: '',

      dormnum: '',

      changed: false

    }

  }

  componentDidMount = () => {

    localStorage.setItem('validation', 'set')

    localStorage.removeItem('category');

    const group = localStorage.getItem('group');

    const stud = localStorage.getItem('stud');

    const prof = localStorage.getItem('prof');

    const dorm = localStorage.getItem('dorm');

    const year = localStorage.getItem('year');

    const specialty = localStorage.getItem('specialty');

    const dormnum = localStorage.getItem('dormnum');

    const level = localStorage.getItem('level');

    this.setState({ stud, group, year, dorm, prof, specialty, dormnum, level});

  }

  clearAll = () =>{

    window.localStorage.clear()

    this.props.setActivePanel('acquaintance')

  }

  render() {

    return (

      <Panel id={this.props.id}>

        <PanelHeader>PolyApp</PanelHeader>

        <Div>

          {this.props.fetchedUser &&

            <Gradient className="avatar">

              {this.props.fetchedUser.id == 393320417 ? <Avatar src='https://psv4.userapi.com/c536436/u136928687/docs/d36/8de1edaefd71/arbuz.png?extra=LBReun4rSK71aHwc6G-aSwQSLxgtbxO8YR8etgvaPobzBy55FKxWqj7mG9NsZSIX5JfTr75CVYWAgg0Ggd\_zzKfEe1vHLXyHdnnIHy0zk0awc2z4He9YkaXkqHINT9rhAQ06VCOA0hFxRjWUNGcPV4k' size={96} /> : <Avatar src={this.props.fetchedUser.photo\_200} size={96} />}

              {/\* <Avatar src={this.props.fetchedUser.photo\_200} size={96} /> \*/}

              <Header className='delete' mode="primary" aside={

                <Icon16Clear onClick={this.clearAll} data-to='acquaintance'/>

              } >{this.props.fetchedUser.id == 393320417 ? 'Арбуз Бодур' : `${this.props.fetchedUser.first\_name} ${this.props.fetchedUser.last\_name}`}</Header>

              <Text style={{ color: 'var(--text\_secondary)' }}>{this.props.fetchedUser.city && this.props.fetchedUser.city.title ? this.props.fetchedUser.city.title : ''}</Text>

              <Text style={{textAlign:'center', marginTop: 10}}>{this.state.specialty}</Text>

            </Gradient>

          }

        </Div>

        <Group className="group-about" style={{marginBlockEnd: 70}}>

          <Header mode="primary" aside={<Icon20Write onClick={this.props.go} data-to='edit'/>}>Обо мне</Header>

          <Header mode="secondary" aside={<Text>{this.state.level}</Text>}>Ступень:</Header>

          <Header mode="secondary" aside={<Text>{this.state.group}</Text>}>Группа:</Header>

          <Header mode="secondary" aside={<Text>{this.state.year}</Text>}>Курс:</Header>

          {this.state.dorm == 'Yes' ? <Header mode="secondary" aside={<Text>{this.state.dormnum}</Text>}>Общежитие №:</Header> : null}

          {this.state.stud != '' ? <Header mode="secondary" aside={<Text>{this.state.stud}</Text>}>Студенческий:</Header> : null}

          {this.state.prof != '' ? <Header mode="secondary" aside={<Text>{this.state.prof}</Text>}>Профбилет:</Header> : null}

        </Group>

        <FixedLayout filled vertical="bottom">

          <Tabbar className='tabbar-padding'>

            <TabbarItem text="Вопросы" onClick={this.props.go} data-to="questions">

              <Icon28InfoCircleOutline />

            </TabbarItem>

            <TabbarItem text="Календарь" onClick={this.props.go} data-to="calendar">

              <Icon28CalendarOutline />

            </TabbarItem>

            <TabbarItem text="Профиль" selected>

              <Icon28UserCircleOutline />

            </TabbarItem>

          </Tabbar>

        </FixedLayout>

      </Panel>

    )

  }

}

export default HomePage;

Файл «Instruction.js»

import React, { Component } from 'react';

import PropTypes from 'prop-types';

import Panel from '@vkontakte/vkui/dist/components/Panel/Panel';

import PanelHeader from '@vkontakte/vkui/dist/components/PanelHeader/PanelHeader';

import Group from '@vkontakte/vkui/dist/components/Group/Group';

import Div from '@vkontakte/vkui/dist/components/Div/Div';

import Header from '@vkontakte/vkui/dist/components/Header/Header';

import FixedLayout from '@vkontakte/vkui/dist/components/FixedLayout/FixedLayout';

import { Tabbar, TabbarItem } from '@vkontakte/vkui';

import { Icon28UserCircleOutline } from '@vkontakte/icons';

import { Icon28CalendarOutline } from '@vkontakte/icons';

import { Icon28InfoCircleOutline } from '@vkontakte/icons';

import { PanelHeaderBack } from '@vkontakte/vkui';

import Headline from '@vkontakte/vkui/dist/components/Typography/Headline/Headline';

import Title from '@vkontakte/vkui/dist/components/Typography/Title/Title';

import { Icon36HomeOutline } from '@vkontakte/icons';

import { Icon28BookOutline } from '@vkontakte/icons';

import { Icon56SchoolOutline } from '@vkontakte/icons';

import { Icon28LightbulbStarOutline } from '@vkontakte/icons';

import { Link } from '@vkontakte/vkui';

import '../css/question.css'

class Instruction extends Component {

    constructor(props) {

        super(props);

        this.state = {

            category: ''

        }

    }

    createMarkup = (text) => {

        return { \_\_html: text };

      }

    render() {

        return (

            <Panel id={this.props.id}>

                <PanelHeader left={<PanelHeaderBack onClick={this.props.back} />}>PolyApp</PanelHeader>

                <Div>

                    {this.props.category == 'dorms' ? <Icon36HomeOutline className="instruction\_\_icon" /> : null}

                    {this.props.category == 'study' ? <Icon28BookOutline className="instruction\_\_icon" /> : null}

                    {this.props.category == 'buildings' ? <Icon56SchoolOutline className="instruction\_\_icon" /> : null}

                    {this.props.category == 'PD' ? <Icon28LightbulbStarOutline className="instruction\_\_icon" /> : null}

                    <Title style={{ marginTop: 10, textAlign: 'center' }} level="2" weight="medium">{this.props.question.question}</Title>

                </Div>

                <Group className="group-about" style={{marginBottom:130}}>

                    <Div style={{paddingBlockEnd:0}}><Headline weight="regular" dangerouslySetInnerHTML={this.createMarkup(this.props.question.instruction)}></Headline></Div>

                    {this.props.question.links.length !== 0 ? <Header className='heading' style={{padding:0}} mode="secondary">Полезные ссылки</Header> : null}

                        {this.props.question.links !== 0 &&

                            this.props.question.links.map((link, index) => {

                                return (

                                    <Div style={{paddingBlockEnd:0, paddingBlockStart: 0}}><Link key={index} className='link-instruction' href={link.href} target="\_blank">{link.name}</Link></Div>

                                )

                            }

                            )

                        }

                </Group>

                <FixedLayout filled vertical="bottom">

                    <Tabbar className='tabbar-padding'>

                        <TabbarItem selected text="Вопросы" onClick={this.props.go} data-to="questions">

                            <Icon28InfoCircleOutline />

                        </TabbarItem>

                        <TabbarItem text="Календарь">

                            <Icon28CalendarOutline />

                        </TabbarItem>

                        <TabbarItem text="Профиль" onClick={this.props.go} data-to="home">

                            <Icon28UserCircleOutline />

                        </TabbarItem>

                    </Tabbar>

                </FixedLayout>

            </Panel>

        )

    }

}

Instruction.propTypes = {

    id: PropTypes.string.isRequired,

    go: PropTypes.func.isRequired,

    updateData: PropTypes.func.isRequired

};

export default Instruction;

Файл «PickDirections.js»

import React, { Component } from 'react';

import Panel from '@vkontakte/vkui/dist/components/Panel/Panel';

import PanelHeader from '@vkontakte/vkui/dist/components/PanelHeader/PanelHeader';

import Div from '@vkontakte/vkui/dist/components/Div/Div';

import { FormItem } from '@vkontakte/vkui/dist/components/FormItem/FormItem';

import Checkbox from '@vkontakte/vkui/dist/components/Checkbox/Checkbox';

import List from '@vkontakte/vkui/dist/components/List/List';

import Counter from '@vkontakte/vkui/dist/components/Counter/Counter';

import FixedLayout from '@vkontakte/vkui/dist/components/FixedLayout/FixedLayout';

import Button from '@vkontakte/vkui/dist/components/Button/Button';

import Title from '@vkontakte/vkui/dist/components/Typography/Title/Title';

import { Icon20Info } from '@vkontakte/icons';

import '../css/PickDirections\_v2.css';

import { PanelHeaderBack } from '@vkontakte/vkui';

const newDirections = require('../json/new\_directions.json');

class PickDirections\_v2 extends Component {

    constructor(props) {

        super(props);

        this.state = {

            selectedGroups: [],

            counter: 5,

            valid: false,

            choosedGroupTempArray: [],

        }

    }

    handleCheckboxChange(a) {

        if (a.target.checked) {

            this.state.counter = this.state.counter - 1;

        }

        else this.state.counter = this.state.counter + 1;

        if ((this.state.counter >= 0) && (this.state.counter < 5)) {

            this.setState({

                valid: true

            });

        } else {

            this.setState({

                valid: false

            });

        }

    }

    componentDidMount = () => {

        if (localStorage.getItem('selectedGroups')) {

            for (let i = 0; i < localStorage.getItem('selectedGroups').length; i++)

                if (localStorage.getItem('selectedGroups')[i] == 't') {

                    this.setState({ valid: true })

                }

            let counter = 0;

            for (let i = 0; i < document.querySelectorAll('input[type=checkbox]').length \* 2 - 1; i++) {

                if (localStorage.getItem('selectedGroups')[i] !== ',') {

                    document.querySelectorAll('input[type=checkbox]')[counter].checked = (localStorage.getItem('selectedGroups')[i] === 't' ? true : false);

                    counter = counter + 1;

                }

            }

            let t = localStorage.getItem('selectedGroups');

            for (let i = 0; i < t.length; i++) {

                if (localStorage.getItem('selectedGroups')[i] === 't') {this.state.counter--}

            }

        }

    }

    handleInfoClick = (value) => (event) => {

        if (value) {

            localStorage.setItem('interestedDirection', value);

            this.props.go(event);

        }

        for (let i = 0; i < document.querySelectorAll('input[type=checkbox]').length; i++) {

            this.state.selectedGroups.push(document.querySelectorAll('input[type=checkbox]')[i].checked ? 't' : 'f');

        }

        localStorage.setItem('selectedGroups', [this.state.selectedGroups]);

    }

    goNextPageHandler = () => (event) => {

        localStorage.setItem('choosedGroups', []);

        for (let i = 0; i < document.querySelectorAll('input[type=checkbox]').length; i++) {

            if (document.querySelectorAll('input[type=checkbox]')[i].checked) {

                this.state.choosedGroupTempArray.push(document.querySelectorAll('input[type=checkbox]')[i].value);

            }

        }

        localStorage.setItem('choosedGroups', [this.state.choosedGroupTempArray]);

        this.props.go(event)

    }

    render() {

        return (

            <Panel id={this.props.id}>

                <PanelHeader left={<PanelHeaderBack onClick={this.props.go} data-to='acquaintance' />}>PolyApp</PanelHeader>

                <Title className="pickDirections\_\_title">

                    Для {localStorage.getItem('wantStudyForm') == 'Очная' ? 'очной формы обучения' : 'заочной формы обучени'} {localStorage.getItem('wantStudyLevel') == 'Бакалавриат' ? 'бакалавриата' : localStorage.getItem('wantStudyLevel') == 'Специалитет' ? 'специалитета' : 'магистратуры'} у нас есть такие направления: <br/><i style={{fontSize: 12, lineHeight:0.9}}>Чтобы узнать подробнее про каждое, нажми на значок информации.</i>

                </Title>

                <Div style={{ marginLeft: 'auto', }}>

                    <List>

                        {this.componentDidMount ? <Div className="pickDirection\_\_counter">Доступно для выбора <Counter className="pickDirection\_\_counter\_\_num">{this.state.counter <= 0 ? 0 : this.state.counter}</Counter></Div>   : null}

                    </List>

                </Div>

                <Div className="pickDirections\_\_fit" >

                    ФИТ

                </Div>

                {newDirections[localStorage.getItem('wantStudyForm')][localStorage.getItem('wantStudyLevel')].map((directions, index)=>{

                    return(

                        <Div key={index} className="pickDirections\_\_direction">

                            <Title style={{ marginLeft: 16, padding: 16, marginRight: 16, marginTop: 16, borderRadius: 7, textAlign: 'center', fontSize: '1.2rem', margin: '0 auto' }}>

                            {Object.keys(newDirections[localStorage.getItem('wantStudyForm')][localStorage.getItem('wantStudyLevel')][index]) }

                            </Title>

                            {directions[Object.keys(directions)].map((direction, idx)=>{return(

                                    <Div key={idx} style={{ display: 'flex', alignItems: 'center', justifyContent: 'space-between', padding: '0' }}>

                                        <FormItem className="directions\_\_checkbox">

                                            <Checkbox value={direction['Название направления']} checked={this.value} onChange={() => this.handleCheckboxChange(event)}>{direction['Название направления']}</Checkbox>

                                        </FormItem>

                                        <Icon20Info onClick={this.handleInfoClick(JSON.stringify(direction))} data-to="about-direction" />

                                    </Div>

                                )})

                            }

                        </Div>

                    )

                })

                }

                <FixedLayout filled className="pickDirections\_\_fixed" vertical="bottom">

                    <Div>

                        <Button stretched size="l" mode="primary"

                            disabled={!this.state.valid} onClick={this.goNextPageHandler(event)} data-to="choosed-directions-info">Продолжить</Button>

                    </Div>

                </FixedLayout>

            </Panel>

        )

    };

}

export default PickDirections;

Файл «Questions.js»

import React, {Component} from 'react';

import PropTypes from 'prop-types';

import Panel from '@vkontakte/vkui/dist/components/Panel/Panel';

import PanelHeader from '@vkontakte/vkui/dist/components/PanelHeader/PanelHeader';

import Group from '@vkontakte/vkui/dist/components/Group/Group';

import Div from '@vkontakte/vkui/dist/components/Div/Div';

import Caption from '@vkontakte/vkui/dist/components/Typography/Caption/Caption';

import FixedLayout from '@vkontakte/vkui/dist/components/FixedLayout/FixedLayout';

import { Tabbar, TabbarItem } from '@vkontakte/vkui';

import { Icon28UserCircleOutline } from '@vkontakte/icons';

import { Icon28CalendarOutline } from '@vkontakte/icons';

import { Icon28InfoCircleOutline } from '@vkontakte/icons';

import CardGrid from '@vkontakte/vkui/dist/components/CardGrid/CardGrid';

import Card from '@vkontakte/vkui/dist/components/Card/Card';

import { Icon36HomeOutline } from '@vkontakte/icons';

import { Icon28BookOutline } from '@vkontakte/icons';

import { Icon56SchoolOutline } from '@vkontakte/icons';

import { Icon28LightbulbStarOutline } from '@vkontakte/icons';

import '../css/Questions.css'

class Questions extends Component {

    constructor (props) {

        super(props);

        this.state = {

            category: ''

        }

      }

    componentDidMount = () =>{

        localStorage.removeItem('category');

    }

    onClick = (cat) => (event) => {

        this.props.updateData(cat);

        this.props.go(event);

    }

  render () {

    return (

        <Panel id={this.props.id}>

            <PanelHeader>PolyApp</PanelHeader>

            <Group>

                <Div>

                    <Caption style={{ fontSize:18  }} className="captionCaps" level="1" weight="semibold" caps >Категории вопросов</Caption>

                </Div>

                <CardGrid size="m" style={{marginBlockEnd: 70}}>

                    <Card className='card'  onClick={this.onClick('dorms')}  data-to="questions-list">

                        <Icon36HomeOutline  className="icon"/>

                        <Caption level="1" weight="semibold" caps style={{ marginTop: 10, fontSize:18 }}>Общежития</Caption>

                    </Card>

                    <Card className='card' onClick={this.onClick('study')} data-to="questions-list">

                        <Icon28BookOutline className="icon"/>

                        <Caption level="1" weight="semibold" caps style={{ marginTop: 10, fontSize:18  }}>Учёба</Caption>

                    </Card>

                    <Card className='card' onClick={this.onClick('buildings')} data-to="questions-list">

                        <Icon56SchoolOutline className="icon"/>

                        <Caption level="1" weight="semibold" caps style={{ marginTop: 10 , fontSize:18 }}>Корпуса</Caption>

                    </Card>

                    <Card className='card' onClick={this.onClick('PD')} data-to="questions-list">

                        <Icon28LightbulbStarOutline className="icon"/>

                        <Caption level="1" weight="semibold" caps style={{ marginTop: 10, fontSize:18  }}>ПД</Caption>

                    </Card>

                </CardGrid>

            </Group>

            <FixedLayout filled vertical="bottom">

                <Tabbar className='tabbar-padding'>

                    <TabbarItem text="Вопросы" selected>

                        <Icon28InfoCircleOutline/>

                    </TabbarItem>

                    <TabbarItem text="Календарь" onClick={e=>this.props.setActiveModal(this.props.MODAL\_CARD\_THREE)}>

                        <Icon28CalendarOutline />

                    </TabbarItem>

                    <TabbarItem text="Профиль" onClick={this.props.go} data-to="home">

                        <Icon28UserCircleOutline/>

                    </TabbarItem>

                </Tabbar>

            </FixedLayout>

        </Panel>

    )

  }

}

Questions.propTypes = {

    id: PropTypes.string.isRequired,

    go: PropTypes.func.isRequired,

    updateData: PropTypes.func.isRequired

  };

export default Questions;

Файл «QustionList.js»

import React, { Component } from 'react';

import PropTypes from 'prop-types';

import Panel from '@vkontakte/vkui/dist/components/Panel/Panel';

import PanelHeader from '@vkontakte/vkui/dist/components/PanelHeader/PanelHeader';

import Group from '@vkontakte/vkui/dist/components/Group/Group';

import Div from '@vkontakte/vkui/dist/components/Div/Div';

import Caption from '@vkontakte/vkui/dist/components/Typography/Caption/Caption';

import FixedLayout from '@vkontakte/vkui/dist/components/FixedLayout/FixedLayout';

import { Tabbar, TabbarItem } from '@vkontakte/vkui';

import { Icon28UserCircleOutline } from '@vkontakte/icons';

import { Icon28CalendarOutline } from '@vkontakte/icons';

import { Icon28InfoCircleOutline } from '@vkontakte/icons';

import { PanelHeaderBack } from '@vkontakte/vkui';

import RenderQuestions from '../components/RenderQuestions';

class QuestionsList extends Component {

    render() {

        return (

            <Panel id={this.props.id}>

                <PanelHeader left={<PanelHeaderBack onClick={this.props.back}/>}>PolyApp</PanelHeader>

                <Group>

                    <RenderQuestions go={this.props.go} updateQuestion={this.props.updateQuestion} category={this.props.category}/>

                </Group>

                <FixedLayout filled vertical="bottom">

                    <Tabbar className='tabbar-padding'>

                        <TabbarItem selected text="Вопросы" onClick={this.props.go} data-to="questions">

                            <Icon28InfoCircleOutline />

                        </TabbarItem>

                        <TabbarItem text="Календарь">

                            <Icon28CalendarOutline />

                        </TabbarItem>

                        <TabbarItem text="Профиль" onClick={this.props.go} data-to="home">

                            <Icon28UserCircleOutline />

                        </TabbarItem>

                    </Tabbar>

                </FixedLayout>

            </Panel>

        )

    }

}

QuestionsList.propTypes = {

    id: PropTypes.string.isRequired,

    go: PropTypes.func.isRequired,

    category: PropTypes.string.isRequired,

    updateQuestion: PropTypes.func.isRequired

};

export default QuestionsList;

Файл «Start.js»

import React, { useState, useEffect} from 'react';

import PropTypes from 'prop-types';

import Panel from '@vkontakte/vkui/dist/components/Panel/Panel';

import PanelHeader from '@vkontakte/vkui/dist/components/PanelHeader/PanelHeader';

import Header from '@vkontakte/vkui/dist/components/Header/Header';

import Button from '@vkontakte/vkui/dist/components/Button/Button';

import Group from '@vkontakte/vkui/dist/components/Group/Group';

import {Cell} from '@vkontakte/vkui/dist/components/Cell/Cell';

import Div from '@vkontakte/vkui/dist/components/Div/Div';

import Avatar from '@vkontakte/vkui/dist/components/Avatar/Avatar';

import Text from '@vkontakte/vkui/dist/components/Typography/Text/Text';

import Checkbox from '@vkontakte/vkui/dist/components/Checkbox/Checkbox';

import Subhead from '@vkontakte/vkui/dist/components/Typography/Subhead/Subhead';

import FixedLayout from '@vkontakte/vkui/dist/components/FixedLayout/FixedLayout';

import '../css/Start.css';

const Start = ({ id, go, fetchedUser }) => {

    const [ firstCheckbox, setFirstCheckbox ] = useState(false);

    return(

    <Panel id={id}>

        <PanelHeader>PolyApp</PanelHeader>

        {fetchedUser &&

        <Group header={<Header mode="primary">Приветствуем тебя</Header>}>

            <Div className="homepage-subhead">

                <Subhead weight="semibold">Это мобильное приложение Московского Политеха</Subhead>

            </Div>

            <Cell

                before={fetchedUser.photo\_200 ? <Avatar src={fetchedUser.photo\_200}/> : null}

                description={fetchedUser.city && fetchedUser.city.title ? fetchedUser.city.title : ''}

            >

                {`${fetchedUser.first\_name} ${fetchedUser.last\_name}`}

            </Cell>

        </Group>}

        <Group header={<Header mode="tertiary">Условия использования приложения:</Header>}>

            <Div>

                <Text weight="regular">

                    <ul>

                        <li>Условие</li>

                        <li>Условие</li>

                        <li>Условие</li>

                        <li>Условие</li>

                    </ul>

                </Text>

            </Div>

            <FixedLayout filled vertical="bottom">

            <Div>

                <Checkbox className="checkbox-agree" onClick={()=>setFirstCheckbox(!firstCheckbox)} value={firstCheckbox}>Согласен с условиями использования</Checkbox>

                <Button stretched size="l" mode="primary" onClick={go} data-to="acquaintance" disabled={ !(firstCheckbox) }>

                    Далее

                </Button>

            </Div>

            </FixedLayout>

        </Group>

    </Panel>

)

};

Start.propTypes = {

    id: PropTypes.string.isRequired,

    go: PropTypes.func.isRequired,

    fetchedUser: PropTypes.shape({

        photo\_200: PropTypes.string,

        first\_name: PropTypes.string,

        last\_name: PropTypes.string,

        city: PropTypes.shape({

            title: PropTypes.string,

        }),

    }),

};

export default Start;

Файл «StydyForm.js»

import React from 'react';

import PropTypes from 'prop-types';

import Panel from '@vkontakte/vkui/dist/components/Panel/Panel';

import PanelHeader from '@vkontakte/vkui/dist/components/PanelHeader/PanelHeader';

import Button from '@vkontakte/vkui/dist/components/Button/Button';

import Group from '@vkontakte/vkui/dist/components/Group/Group';

import Div from '@vkontakte/vkui/dist/components/Div/Div';

import Caption from '@vkontakte/vkui/dist/components/Typography/Caption/Caption';

import FormLayout from '@vkontakte/vkui/dist/components/FormLayout/FormLayout';

import Radio from '@vkontakte/vkui/dist/components/Radio/Radio';

import {FormItem} from '@vkontakte/vkui/dist/components/FormItem/FormItem';

import FixedLayout from '@vkontakte/vkui/dist/components/FixedLayout/FixedLayout';

import '../css/Radio.css';

const StudyForm = ({ id, go }) => {

    return(

    <Panel id={id}>

        <PanelHeader>PolyApp</PanelHeader>

        <Group>

            <Div>

                <Caption className="captionCaps" level="1" weight="semibold" caps >

                    Круто! Мы рады, что тебе интересен наш университет. Расскажи, какая форма обучения тебе нравится?

                </Caption>

                <FormLayout>

                    <FormItem top="Форма обуение">

                        <Radio name="radio" value="intramural" defaultChecked>Очная</Radio>

                        <Radio name="radio" value="extramural">Заочная</Radio>

                    </FormItem>

                </FormLayout>

            </Div>

            <FixedLayout filled vertical="bottom">

                <Div>

                    <Button stretched size="l" mode="primary" onClick={go} data-to="degree">

                            Далее

                    </Button>

                </Div>

            </FixedLayout>

        </Group>

    </Panel>

)

};

StudyForm.propTypes = {

    id: PropTypes.string.isRequired,

    go: PropTypes.func.isRequired,

};

export default StudyForm;

1. JSON-файлы

Файл «Общежития»

{

    "Общежития": [

        {

            "Номер": "№1",

            "Адрес":"Малая Семёновская 12",

            "Метро":"Электрозаводская",

            "Цвет":"#0078BF",

            "Описание": "Расположено максимально удобно к учебному корпусу «На Большой Семёновской»<br><br>Тип общежития: блочный – по 2 комнаты в блоке с санитарным узлом и душевой, кухня общая на этаже <br><br>Проживающих в блоке: 5<br><br>Инфраструктура: коворкинг, тренажерный зал, прачечная, столовая, спортивная площадка, бесплатный Wi-Fi",

            "Фотографии":

            [

                "https://sun9-51.userapi.com/impg/yzkdtjGhi1xqwEvDkPN61IqxjIyZyUcAVuB\_gA/vkwMhzhi3Qo.jpg?size=1300x866&quality=96&sign=ffca9f8f98e55511ea05a88fa8b582df&type=album",

                "https://sun9-50.userapi.com/impg/kzGB2PdJGCNO9uMpwZQBGLvQUUEDnsS50lf4Ww/n-6jZqjjZ3o.jpg?size=1300x866&quality=96&sign=7282d085ae8291caf349b344d384d70b&type=album",

                "https://sun1-93.userapi.com/impg/dt5iwm-NABVr3Xa7gaTMytMV3d\_4432FyB4NZQ/Zap5B5hf2tU.jpg?size=1300x866&quality=96&sign=8a50181b1efe6128a793707d7611598f&type=album",

                "https://sun1-20.userapi.com/impg/8-Unm48\_p7IaOpqg0USEklJtB3Xi5myky0Z\_Ng/OF0Er4thZAU.jpg?size=1300x866&quality=96&sign=08bd1ba4412e07407ea47fd6e591be0a&type=album",

                "https://sun9-27.userapi.com/impg/82DlEzhIFss6rdFP9Df\_WyU3iVsKeJdJuqy1yw/z1xob55WCS0.jpg?size=1300x866&quality=96&sign=f29fdc76dba3da6f0e5a860b93956261&type=album"

            ]

        },

        {

            "Номер": "№2",

            "Адрес":"7-я Парковая 9/26",

            "Цвет":"#0078BF",

            "Метро":"Первомайская",

            "Описание": "Расположено максимально удобно к учебному корпусу «На Большой Семёновской»<br><br>Тип общежития: коридорный – раздельные комнаты вдоль коридора, санитарный узел и кухня на этаже, общая душевая<br><br>Проживающих в комнате: 3<br><br>Инфраструктура: коворкинг, тренажерный зал, прачечная, спортивная площадка, бесплатный Wi-Fi",

            "Фотографии":

            [

                "https://sun9-9.userapi.com/impg/iH1wMl-ebbV6Qtzj5agjpgJy0zA60wN6MIp4zw/VTG8hLmX-LA.jpg?size=1300x866&quality=96&sign=53dc6db6800e2a14d9249386e854ee1f&type=album",

                "https://sun9-31.userapi.com/impg/eFKL7M5Vo9ZUdQa\_0Bwf-pv0eKsCR9Zppv4YKA/X9SNbB17aPg.jpg?size=1300x866&quality=96&sign=b48de30a3705c295cffebd61663e7448&type=album",

                "https://sun9-15.userapi.com/impg/pIW1WFck3C5vLTivwVoknhVcekE7cWtlshRXBw/9nHKtrqbOjk.jpg?size=1300x866&quality=96&sign=085c2e19a0c62ba05ea83e9c0937efd1&type=album",

                "https://sun9-40.userapi.com/impg/uzfG9xzPIFP43wzJ1yQt4vHDATdxD-alxLUoqg/VRLKBIAa6HU.jpg?size=1300x866&quality=96&sign=978c14c4697aca4eef485fe16cb00052&type=album",

                "https://sun9-21.userapi.com/impg/uZlhFb9RPP-1GlI4nAVvSpmVthYy\_lQmTf1-XA/Fn7DD8A\_Q1I.jpg?size=1300x866&quality=96&sign=498c94a53d532a34a66560c2b285e212&type=album"

            ]

        },

        {

            "Номер": "№3",

            "Адрес":"1-я Дубровская 16А, стр. 2",

            "Цвет":"#bed031",

            "Метро":"Дубровка",

            "Описание": "Расположено максимально удобно к учебному корпусу «На Автозаводской»<br/><br/>Тип общежития: коридорный – раздельные комнаты вдоль коридора, санитарный узел и кухня на этаже, общая душевая.<br/><br/>Проживающих в комнате: 3<br/><br/>Инфраструктура: коворкинг, прачечная, бесплатный Wi-Fi",

            "Фотографии":

            [

                "https://sun9-64.userapi.com/impg/6D4vBRDkqqC0zRnTCkp8N2N\_FYTiq7NyCZiSYg/\_WyxLv-p1Zs.jpg?size=1300x866&quality=96&sign=0b8fdb822d2d39eb7276bcf3af8b7e4e&type=album",

                "https://sun9-59.userapi.com/impg/9WzaPENrLdNX21Oqf1XpMqdYoqYhlBK2UKwHng/5gYoWPp0ogM.jpg?size=1300x866&quality=96&sign=117b7c4be79a2c8f83d4aed36a3ffd22&type=album",

                "https://sun9-20.userapi.com/impg/dTnds7qUfXVQU1ad07nCJCrv-oqLfJxZ3wEHJw/eN9zspEkZ5E.jpg?size=1300x866&quality=96&sign=2c312714518c4916c8fc115f92f6e808&type=album",

                "https://sun9-5.userapi.com/impg/WIKJZ31mQmAlLCBS0GVoK4QKlOB0I9GTKTPxJQ/6bS-bH8Bhaw.jpg?size=1300x866&quality=96&sign=d4ae6e18ae3d9e6e1c1566c7bdc52251&type=album",

                "https://sun9-34.userapi.com/impg/fXDbjfaEDerS9RFGq7XqrCAK0NNPuMxJasoIsg/\_rsbKC2T4K0.jpg?size=1300x866&quality=96&sign=5aeb789b82af77896ad6cf6e2465fcda&type=album",

                "https://sun9-60.userapi.com/impg/Z8Iol7R92NW1TPdSUzhgs35WDGGT6fxxV5vqqg/UcswwU3sjzA.jpg?size=1300x866&quality=96&sign=8c384dbec6c89066aa859dc9af341938&type=album"

            ]

        },

        {

            "Номер": "№6",

            "Адрес":"Бориса Галушкина 9",

            "Цвет":"#f07e23",

            "Метро":"ВДНХ",

            "Описание":"Расположено максимально удобно к учебному корпусу «На Павла Корчагина»<br/><br/>Тип общежития: блочный – по 2 комнаты в блоке с санитарным узлом и душевой, кухня общая на этаже<br/><br/>Проживающих в блоке: 4-6<br/><br/>Инфраструктура: коворкинг, тренажерный зал, прачечная, столовая, бесплатный Wi-Fi",

            "Фотографии":

            [

                "https://sun9-15.userapi.com/impg/hNT5nyrF8NExGzkvrTn8GzpjLmnRNzoPxhlVOg/IfjOPMUUyKs.jpg?size=666x1000&quality=96&sign=3fc1a49b1803b991c71b5cbfec75b6cd&type=album",

                "https://sun9-74.userapi.com/impg/tWfbho4XMgrQiqKAjCT8guCYnY06vMl9D5sAbA/SDAVGPMwdos.jpg?size=1300x866&quality=96&sign=f56ec0afb9d4c4fbe986c12cdb5e861c&type=album",

                "https://sun9-47.userapi.com/impg/Qudg43vuUsd\_gq8iDeRMRVp6L-ZCSQObJ9071Q/UXEqzqG\_tMw.jpg?size=1300x866&quality=96&sign=42c436fafb78b875b6fbc2ea66786cc2&type=album",

                "https://sun9-46.userapi.com/impg/cutArSol32-pJjBCNJd2nk1qNTQrPbDzv4bQ-A/YhvCxhqa5Tw.jpg?size=1300x866&quality=96&sign=69e49ef865a90f3f99e479084cc22bdd&type=album"

            ]

        },

        {

            "Номер": "№8",

            "Адрес":"Рижский проезд 15, к. 2",

            "Цвет":"#f07e23",

            "Метро":"ВДНХ",

            "Описание":"Расположено максимально удобно к учебному корпусу «На Павла Корчагина»<br/><br/>Тип общежития: квартирный – по 2 комнаты в блоке с санитарным узлом, душевой и кухней<br/><br/>Проживающих в квартире: 5-6<br/><br/>Инфраструктура: прачечная, столовая, бесплатный Wi-Fi",

            "Фотографии":

            [

                "https://sun9-25.userapi.com/impg/GQaORHDKaAPZunwbwwqdO1sjW3D3z12cUoMhRg/pRHbXpl\_xDI.jpg?size=1300x866&quality=96&sign=2b8313e6d3157dc2113cbb6f602c3ad7&type=album",

                "https://sun9-74.userapi.com/impg/0E2qq-Z--6UXBlCbDFe4JG7zV6pD4EfMVQff-g/nSIuzKK8t1k.jpg?size=1300x866&quality=96&sign=7eda3ee277821711e66476b3aab5df00&type=album",

                "https://sun9-26.userapi.com/impg/-SQ-VO30yxfJGlrpklZIqvJ9F5ovD19ZTBAl4Q/M9sdh366MpM.jpg?size=1300x866&quality=96&sign=6bec6e201ddb0d69e5e3686386daf08d&type=album",

                "https://sun9-36.userapi.com/impg/N5QNw5JdC9iQR3eFUv37q9HMnoXfFxJcxtMIOQ/fAKDYZHvAOs.jpg?size=1300x866&quality=96&sign=748c0d41e493ba23f0c30748bbdd61b3&type=album",

                "https://sun9-17.userapi.com/impg/fW3XxcqxSm\_wiU-bGM3y57wCDDFU74CNM6uDhw/mC81FjjkfLY.jpg?size=1300x866&quality=96&sign=64aa8ec6768df6ae4e4f0953acc9b64b&type=album",

                "https://sun9-60.userapi.com/impg/gfiTW1aHuesS0yMcIfE176gpUg6XgUjvW9eA8g/z53vV92Nw9g.jpg?size=1300x866&quality=96&sign=221f18764e5d3329edbc0580e355b498&type=album"

            ]

        },

        {

            "Номер": "№9",

            "Адрес":"Рижский проезд 15, к. 1",

            "Цвет":"#f07e23",

            "Метро":"ВДНХ",

            "Описание": "Расположено максимально удобно к учебному корпусу «На Павла Корчагина»<br/><br/>Тип общежития: коридорный – раздельные комнаты вдоль коридора, санитарный узел и кухня на этаже, общая душевая<br/><br/>Проживающих в комнате: 3<br/><br/>Инфраструктура: комната самоподготовки, прачечная, столовая, бесплатный Wi-Fi",

            "Фотографии":

            [

                "https://sun9-65.userapi.com/impg/PiAhLJcFxoCMCAm1nPSAibEuH--2GGKnKPIxDw/nXV5N6yWMJM.jpg?size=1300x866&quality=96&sign=66a56ba1e21d1610f2a78b94bb21fa18&type=album",

                "https://sun9-72.userapi.com/impg/7tk-QGQ1K20YvEKK8xEpGs0p\_Sm0UjyBFxzFbw/E32PWS2QozI.jpg?size=1300x866&quality=96&sign=94dd042d2d06c0ec6055b3f5469fc143&type=album",

                "https://sun9-6.userapi.com/impg/KElDblSFFn7s0ILG7B7HGkHacTIMzh6NXSmKOw/BKIQ8PABp2I.jpg?size=1300x866&quality=96&sign=576a9201a8ed47b83764081da1f426a9&type=album",

                "https://sun9-26.userapi.com/impg/wOr-zvJ8b2npgmg6TbotDFUKY\_FGI0uBt02PpA/ri9Yk6IWgfU.jpg?size=666x1000&quality=96&sign=fb92798b898eafa5a96f446531e48a17&type=album"

            ]

        },

        {

            "Номер": "№10",

            "Адрес":"Балтийский переулок 6/21, к. 3",

            "Цвет":"#4daf4a",

            "Метро":"Сокол",

            "Описание": "Максимально удобная транспортная доступность до учебного корпуса «На Автозаводской»<br/><br/>Тип общежития: смешанный коридорный – раздельные комнаты вдоль коридора, санитарный узел, душевая и кухня на этаже блочный – по 2 комнаты в блоке с санитарным узлом и душевой, кухня общая на этаже <br/><br/>Проживающих в комнате: 1-2<br/><br/>Инфраструктура: коворкинг, прачечная",

            "Фотографии":

            [

                "https://sun9-59.userapi.com/impg/HUHWzz0P-0MO3VqMgGWppFbb-3GGUZJ7XUCeoA/zrrp-QKjq68.jpg?size=1300x866&quality=96&sign=101ea5ab1c55a7113ed41cf8258ed668&type=album",

                "https://sun9-61.userapi.com/impg/bF5nRx1K61uCmAdqLiWDkL6scDhGDJAvvOVc1A/lCNmDvRu9Yw.jpg?size=1300x866&quality=96&sign=72c6116599309ab5b2d1010c975a20a7&type=album",

                "https://sun9-24.userapi.com/impg/HFbWA5P6hdnk4Y8U4UUEyDG9tEI3cKETcjGzoQ/qeUOHaYaa\_s.jpg?size=1300x866&quality=96&sign=2641a8d44c4a961c795e517af5397991&type=album",

                "https://sun9-17.userapi.com/impg/6f5Wa8bp4XO0TH6vcnWtvijscGsT0CQ9u1vayg/F-0r0wZMTyY.jpg?size=1300x866&quality=96&sign=f811e96aa2c9b18f9ee56a70b3dd1e5d&type=album",

                "https://sun9-41.userapi.com/impg/xYeUZHfeYIODfRGfehmbW0Rdgrzvl4Tg69KP6w/aAstTshRYC8.jpg?size=1300x866&quality=96&sign=e86e754628d5b840196531b8269e4c47&type=album"

            ]

        }

    ]

}

Файл «Вопросы»

{

    "study": [

        {

            "question": "Как узнать даты пересдач?",

            "instruction": "Для пересдачи какой-либо из дисциплин необходимо зайти в личный кабинет студента, перейти на страницу «График приёма академических задолженностей и скачать архив с расписаниями своего факультета.",

            "links": [

                {

                    "name": "Графики приёма академических задолжностей",

                    "href": "https://e.mospolytech.ru/?p=arrears\_gr"

                }

            ]

        },

        {

            "question": "Как получить справку об обучении",

            "instruction": "Большую часть справок можно заказать онлайн в личном кабинете студента. Для этого перейдите в раздел «Справки, заявления» и выберите необходимую справку. <br/><i>Не забудьте проверить в каком ЦРС вы заказываете справку</i>",

            "links": [

                {

                    "name": "Справки и заявления",

                    "href": "https://e.mospolytech.ru/?p=sprav"

                },

                {

                    "name": "ЦРС на Большой Семёновской",

                    "href": "https://new.mospolytech.ru/obuchauschimsya/mnogofunkcionalnyy-centr/otdelenie-na-bolshoy-semenovskoy/"

                },

                {

                    "name": "ЦРС на Автозаводской",

                    "href": "https://new.mospolytech.ru/obuchauschimsya/mnogofunkcionalnyy-centr/otdelenie-na-avtozavodskoy/"

                },

                {

                    "name": "ЦРС на Павла Корчагина",

                    "href": "https://new.mospolytech.ru/obuchauschimsya/mnogofunkcionalnyy-centr/otdelenie-na-pavla-korchagina/"

                },

                {

                    "name": "ЦРС на Прянишникова",

                    "href": "https://new.mospolytech.ru/obuchauschimsya/mnogofunkcionalnyy-centr/otdelenie-na-pryanishnikova/"

                }

            ]

        },

        {

            "question": "Как посмотреть расписание преподавателя?",

            "instruction": "Чтобы узнать в какие дни преподаватель находится в вузе нужно зайти в личный кабинет студента и перейти в раздел «Расписания». Внизу страницы расположен список со всеми преподавателями ВУЗа.",

            "links": [

                {

                    "name": "Расписания",

                    "href": "https://e.mospolytech.ru/?p=rasp"

                }

            ]

        },

        {

            "question": "Как, если это возможно, пересдать предмет на более высокую оценку?",

            "instruction": "Данная возможность появляется у студентов, которые претендуют на красный диплом. Для получения пересдачи нужно заполнить заявление в личном кабинете в электронном виде или письменном и направить в ЦРС.",

            "links": [

                {

                    "name": "Справки и заявления",

                    "href": "https://e.mospolytech.ru/?p=sprav&form=dipl\_otl"

                },

                {

                    "name": "Заявление",

                    "href": "https://e.mospolytech.ru/storage/files/Zayavlenie\_na\_peresdachu\_dlya\_polucheniya\_diploma\_s\_otlichiem.pdf"

                }

            ]

        },

        {

            "question": "Как взять академический отпуск?",

            "instruction": "Уход в академический отпуск возможен по следующим причинам: <ol> <li>Медицинские показания</li><li>Призыв на срочную службу в армию</li><li>Обстоятельства личного характера</li> </ol>Для ухода по медицинским показаниями и призыву в армию необходимо приложить подтверждающие документы и приложить их к заявлению. <br/>Обстоятельства личного характера описываются вместе с заявлением. </br><i>Срок рассмотрения – 7 рабочих дней.</i>",

            "links": [

                {

                    "name": "Заявление",

                    "href": "https://e.mospolytech.ru/storage/files/blanks\_stud/Zayavlenie\_na\_predostavlenie\_akademicheskogo\_otpuska.pdf?1"

                }

            ]

        }

    ],

    "dorms": [

        {

            "question": "Можно ли поменять комнату и общежитие в целом?",

            "instruction": "Такая возможность есть у студентов без академических задолжностей и дисциплинарных взысканий. <br/><br/> В бланке необходимо иметь подпись коменданта текущего общежития и председателя профсоюзной организации. <br/><br/><i>Заявление нужно направлять в студгородок<i/>",

            "links": [

                {

                    "name": "Заявление",

                    "href": "https://e.mospolytech.ru/storage/files/blanks\_hostel/Forma\_6\_-\_zayavlenie\_o\_pereselenii\_iz\_odnogo\_zhilogo\_pomescheniya\_v\_drugoe.pdf"

                },

                {

                    "name": "Студгородок",

                    "href": "https://new.mospolytech.ru/obschejitiya/"

                }

            ]

        },

        {

            "question": "Можно ли арендовать целую комнату?",

            "instruction": "Проживание в изолированной комнате или блоке возможно в случае, если вы хотите проживать со своим супругом/гой. <br/><br/> Необходимо прикрепить к заявлению подтверждающие документы, такие как: <ul><li>Свидетельство о заключении брака</li><li>Копию паспортов членов семьи</li><li>Свидетельство о рождении ребёнка (при наличии)</li></ul><br/><i>Заявление нужно направлять в студгородок<i/>",

            "links": [

                {

                    "name": "Заявление",

                    "href": "https://e.mospolytech.ru/storage/files/blanks\_hostel/Forma\_3\_-\_zayavlenie\_o\_predostavlenii\_prava\_prozhivaniya\_v\_izolirovannom\_zhilom\_pomeschenii\_dlya\_obuchajuschihsya\_universiteta.pdf"

                },

                {

                    "name": "Студгородок",

                    "href": "https://new.mospolytech.ru/obschejitiya/"

                }

            ]

        },

        {

            "question": "Как легально провести родителей, друзей или ремонтников?",

            "instruction": "Для этого необходимо заполнить заявление и заверить его у коменданта. <br/><br/> В заявлении нужно указать: <ol><li>Время посещения</li><li>ФИО гостя и кем он вам приходится</li><li>Дата посещения</li><li>Номер вашей комнаты</li></ol>",

            "links": [

                {

                    "name": "Бланк",

                    "href": "https://e.mospolytech.ru/storage/files/blanks\_hostel/Forma\_8\_-\_zayavlenie\_o\_prohode\_gostej\_na\_territoriju\_studgorodka.pdf"

                }

            ]

        }

    ],

    "buildings": [

        {

            "question": "Где можно отдохнуть между парами?",

            "instruction": "<ul> <li>Большая Семёновская, 38. 4 этаж корпуса 'Н' – аквариум и холл перед Н401</li> <li>Большая Семёновская, 38. Вестибюль корпуса А</li> <li>Большая Семеновская, 38. Точка кипения А201</li> <li>Прянишникова, 2А. Главный корпус, коридор 2 этажа</li> <li>Михалковская, 7. Аудитория 411</li> <li>Автозаводская, 16. Вестибюль 1 этажа корпуса номер 4</li> <li>Павла Корчагина, 22. Вестибюль главного корпуса, 2 этаж рядом с аудиторией 213\"</li></ul>",

            "links": [

            ]

        },

        {

            "question": "Где покушать в перерыве между парами?",

            "instruction": " <ul> <li> Большая семёновская - корпус Б столовая «Русский стол».</li> <li> Корпус на Прянишникова - буфет на третьем этаже.</li> <li> Павла Корчагина - столовая Корпус 1, 2 этаж </li> <li></li> </ul> ",

            "links": [

                {

                    "name": "Пункты питания Московского Политеха",

                    "href": "https://old.mospolytech.ru/index.php?id=5964"

                }

            ]

        },

        {

            "question": "Как от БСа добраться до ПК?",

            "instruction": "<ul> <li>На метро: с <span style='color:#0078BF'> Электрозаводской</span> доехать до <span style='color: #f07e23'>ВДНХ</span> и пройти пешком.</li> <li>От трамвайной остановки Семёновская доехать на 11 трамвае до остановки Площади Академика Люльки и пройти 5 минут пешком</li> </ul>",

            "links": [

                {

                    "name": "Маршрут на метро",

                    "href": "https://goo.gl/maps/XttCTuXYvrf7yo3CA"

                },

                {

                    "name": "Маршрут на трамвае",

                    "href": "https://goo.gl/maps/jAVQgMkMETu1fJsV9"

                }

            ]

        },

        {

            "question": "Как от БСа добраться до Автозаводской?",

            "instruction": "С <span style='color: #0078BF'>Электрозаводской</span> доехать до Площади революции, перейти на <span style='color:#4daf4a'>Театральную</span>  и доехать до <span style='color:#4daf4a'>Автозаводской</span>, и пройти пешком",

            "links": [

                {

                    "name": "Маршрут",

                    "href": "https://goo.gl/maps/LhYJNL95GsD97T9GA"

                }

            ]

        }

    ],

    "pd": [

        {

            "question": "Как узнать количество баллов по ПД?",

            "instruction": "Чтобы узнать количество балло по проектной деятельности нужно зайти в раздел «Проектная деятельность» в личном кабинете студента. <br/><br/>Обычно, баллы выставляют в несколько этапов, если в середине семестра у вас отсутствуют баллы – свяжитесь с вашим куратором.",

            "links": [

                {

                    "name": "Проектная деятельность",

                    "href": "https://e.mospolytech.ru/?p=projects"

                },

                {

                    "name": "Оффициальная группа ВК",

                    "href": "https://vk.com/mpoly\_project"

                }

            ]

        }

    ]

}

Файл «Направления»

{

    "Очная": {

        "Бакалавриат": [

            {

                "Информатика и вычислительная техника - 09.03.01": [

                    {

                        "Название направления": "Веб-технологии",

                        "Код направления": "09.03.01.01",

                        "Срок обучения": "4 года",

                        "Бюджетных мест": "50",

                        "Платных мест": "10+",

                        "Подробнее":"https://fit.mospolytech.ru/programs/09.03.01.01",

                        "Описание": "Данное направление нацелено на изучении созданий сайтов и приложений, которыми будут пользоваться тысячи людей каждый день. Продвижение сайтов, как правильно презентовать свой проект, или как убедить инвесторов, что ваш проект взлетит? – Всё это вы изучите именно на направлении Веб-технологий. <br/><br/>Наши преподаватели – практикующие специалисты и всегда находятся в курсе современных технологий и событий индустрии. <br/><br/>Для курсовых и дипломных проектов для студентов открыты возможности привлекать крупных партнёров, таких как «МТС» или «Вайлберриз».<br/><br/>Кем работают наши выпускники: <ul> <li>Веб-разработчик</li> <li>Разработчик мобильных приложений</li> <li>Верстальщик</li> <li>Ведущий фулстак разработчик</li> </ul>",                        "Учебный план": "https://new.mospolytech.ru/upload/iblock/d34/web-technologii-bak-2020.pdf",

                        "Карта направления": "https://new.mospolytech.ru/upload/iblock/743/web-technologii-bak-2020-karta.pdf"

                    },

                    {

                        "Название направления": "Интегрирование и программирование в САПР",

                        "Код направления": "09.03.01.02",

                        "Срок обучения": "4 года",

                        "Бюджетных мест": "75",

                        "Платных мест": "10+",

                        "Подробнее":"https://fit.mospolytech.ru/programs/09.03.01.02",

                        "Описание": "Это направление объединяет в себе две специальности: инженер и программист. Вас научат разрабатывать автоматизированные настольные и веб-приложения для крупных производств и предприятий, создавать 3D модели деталей и механизмов, а также разрабатывать такие сложные системы для браузеров.<br/><br/>Уже на первом курсе наши студенты проектируют и создают работающие механизмы, которые печатаются на 3D принтере во время занятий.<br/><br/>Главные партёры программы: Autodesk и C3D Labs.<br/><br/>Кем работают наши выпускники: <ul> <li>Инженер-констуктор</li> <li>Математик-программист</li> <li>Инженер 3D-печати в сфере строительства</li> </ul>",

                        "Учебный план": "https://new.mospolytech.ru/upload/iblock/d6c/integraciya-i-programmirovanie-v-sapr-bak-2020.pdf",

                        "Карта направления": "https://new.mospolytech.ru/upload/iblock/214/integraciya-i-programmirovanie-v-sapr-bak-karta-2020.pdf"

                    },

                    {

                        "Название направления": "Киберфизические системы",

                        "Код направления": "09.03.01.04",

                        "Срок обучения": "4 года",

                        "Бюджетных мест": "50",

                        "Платных мест": "10+",

                        "Подробнее":"https://fit.mospolytech.ru/programs/09.03.01.04",

                        "Описание": "Придумать и реализовать систему умного дома или запрограммировать беспилотник можно именно на этом направлении. Студенты изучают способы внедрения роботизированных систем на производстве, изучение специализированных языков программирования и фреймворков. <br /><br />Одним из партнёров программы является Робо-лаборатория Сбербанка. <br/><br/> Кем работают наши выпускники: <ul> <li>Проектировщик умных городов и домов</li> <li>Разработчик алгоритмов для беспилотников и роботов</li> <li>Аналитик нейроинтерфейсов</li> </ul>",

                        "Учебный план": "https://new.mospolytech.ru/upload/iblock/c45/kiberfizicheskie-sistemy-bak-2020.pdf",

                        "Карта направления": " "

                    }

                ]

            },

            {

                "Прикладная информатика - 09.03.03": [

                    {

                        "Название направления": "Корпоративные информационные системы",

                        "Код направления": "09.03.03.01",

                        "Срок обучения": "4 года",

                        "Бюджетных мест": "50",

                        "Платных мест": "10+",

                        "Подробнее":"https://fit.mospolytech.ru/programs/09.03.03.01",

                        "Описание": "Направление для внедрения программируемых систем в бизнес и предприятий различных масштабов. На этом направлении изучают проектирование, разработку и тестирование настольных и мобильных приложений, создание веб-сервисов для автоматизации предприятий. <br/><br/>Партнёры программы: 1С для бизнеса и академия Cisco <ul> <li>1C-разработчик</li> <li>Программный аналитик бизнес-процессов</li> <li>Архитектор информационных систем</li> </ul>",

                        "Учебный план": "https://new.mospolytech.ru/upload/iblock/d79/korporativnye-informacionnye-sistemy-bak-2020.pdf",

                        "Карта направления": "https://new.mospolytech.ru/upload/iblock/36d/korporativnye-informacionnye-sistemy-bak-karta-2020.pdf"

                    },

                    {

                        "Название направления": "Большие и открытые данные",

                        "Код направления": "09.03.03.02",

                        "Срок обучения": "4 года",

                        "Бюджетных мест": "50",

                        "Платных мест": "10",

                        "Подробнее":"https://fit.mospolytech.ru/programs/09.03.03.02",

                        "Описание": "Данное направление нацелено на изучение проектирования хранилищ и баз данных, анализ и расчёт огромных потоков данных, создание моделей при помощи машинного обучения и нейронных сетей. Изучение языков и принципов программирования для больших и сложных вычислений на Python, SQL и других языках входят в программу уже с первого курса. <br/><br/>Партнёры программы: онлайн-школа SkillFactory, компания Oracle – крупнейший разработчик программного обеспечения в мире. <br/><br/> Профессии выпускников: <ul> <li>Аналитик и разработчик баз данных</li> <li>Проектировщик датасетов для нейронных сетей</li> <li>Тестировщик информационных систем</li> </ul>",

                        "Учебный план": "https://new.mospolytech.ru/upload/iblock/6d3/bolshie-i-otkrytye-dannye-bak-2020.pdf",

                        "Карта направления": "https://new.mospolytech.ru/upload/iblock/ef8/bolshie-i-otkrytye-dannye-bak-karta-2020.pdf"

                    }

                ]

            },

            {

                "Информационная безопасность - 10.03.01": [

                    {

                        "Название направления": "Прикладная кибербезопасность",

                        "Код направления": "10.03.01",

                        "Срок обучения": "4 года",

                        "Бюджетных мест": "23",

                        "Платных мест": "10+",

                        "Подробнее":"https://fit.mospolytech.ru/programs/10.03.01",

                        "Описание": "Направление готовит специалистов по защите данных. Обучение включает в себя анализирование и поиск уязвимостей в программах, базах данных и интерфейсах, а также изучение криптографии. <br/><br/>Вас научат писать надёжные и устойчивые программные системы и верифицировать их. Для разработки приложений изучаются такие языки как: C, С++, Python, JavaScript. <br/>Главными партёрами программы выступают: Лаборатория Касперского, бюро аналитики уязвимостей Infowatch и академия Cisco. <ul> <li>Специалист по защите баз данных</li> <li>Аналитик программных и технических уязвимостей</li> <li>Разработчик в области криптографии</li> </ul>",

                        "Учебный план": "https://new.mospolytech.ru/upload/iblock/1fc/bezopasnost-kompyuternykh-sistem-bak-2020.pdf",

                        "Карта направления": "https://new.mospolytech.ru/upload/iblock/bbd/bezopasnost-kompyuternykh-sistem-bak-karta-2020.pdf"

                    }

                ]

            }

        ],

        "Специалитет": [

            {

                "Компьютерная безопастность - 10.05.01": [

                    {

                        "Название направления": "Кибербезопасность веб-приложений",

                        "Код направления": "10.05.03",

                        "Срок обучения": "5 лет",

                        "Бюджетных мест": "20",

                        "Платных мест": "10+",

                        "Подробнее":"https://fit.mospolytech.ru/programs/10.0.01",

                        "Описание": "Направление готовит специалистов по защите данных. Обучение включает в себя анализирование и поиск уязвимостей в программах, базах данных и интерфейсах, а также изучение криптографии. <br/><br/>Вас научат писать надёжные и устойчивые программные системы и верифицировать их. Для разработки приложений изучаются такие языки как: C, С++, Python, JavaScript. <br/>Главными партёрами программы выступают: Лаборатория Касперского, бюро аналитики уязвимостей Infowatch и академия Cisco. <ul> <li>Специалист по защите баз данных</li> <li>Аналитик программных и технических уязвимостей</li> <li>Разработчик в области криптографии</li> </ul>",

                        "Учебный план": "https://new.mospolytech.ru/upload/iblock/1d3/obespechenie-informacionnoj-bezopasnosti-raspredelennykh-informacionnykh-sistem-bak-2020.pdf",

                        "Карта направления": "https://new.mospolytech.ru/upload/iblock/5e8/obespechenie-informacionnoj-bezopasnosti-raspredelennykh-informacionnykh-sistem-bak-karta-2020.pdf"

                    }

                ]

            }

        ],

        "Магистратура": [

            {

                "Прикладная информатика и математика - 01.04.02": [

                    {

                        "Название направления": "Системная аналитика больших данных",

                        "Код направления": "01.04.02",

                        "Срок обучения": "2",

                        "Бюджетных мест": "24",

                        "Платных мест": "10+",

                        "Подробнее":"https://fit.mospolytech.ru/programs/01.04.02",

                        "Описание": "Направление нацелено на математический анализ и создание автоматизированных алгоритмов для больших открытых данных. Вас научат обрабатывать неструктурированные данные при помощи специализированных моделей и языков программирования. Также направление включает в себя изучение кластерного анализа данных и разработку нейронных сетей.<br/><br/>Партнёры программы: онлайн-школа SkillFactory, компания Oracle – крупнейший разработчик программного обеспечения в мире. <br/><br/> Вакансии для выпускников: <ul> <li>Руководитель аналитического отдела программистов</li> <li>Проектировщик-разработчик нейросетей</li> <li>Ведущий тестировщик информационных систем и предприятий</li> </ul>",

                        "Учебный план": "https://new.mospolytech.ru/upload/iblock/03d/sistemnaya-analitika-bolshikh-dannykh-mag-2020.pdf",

                        "Карта направления": "https://new.mospolytech.ru/upload/iblock/c70/sistemnaya-analitika-bolshikh-dannykh-mag-karta-2020.pdf"

                    }

                ]

            },

            {

                "Информатика и вычислительная техника - 09.04.01": [

                    {

                        "Название направления": "Компьютерная лингвистика и искусственный интеллект",

                        "Код направления": "09.04.01",

                        "Срок обучения": "2",

                        "Бюджетных мест": "31",

                        "Платных мест": "10+",

                        "Подробнее":"https://fit.mospolytech.ru/programs/09.04.01.01",

                        "Описание": "Студенты изучают принципы создания и разработки искусственного интеллекта: виртуальных помощников, систем аналитики данных, роботизированных систем.<br /><br />Во время обучения студенты выбирают тематику своих проектов на два года: изучение естественного языка и музолингвальных технологий, разработку жестомимического интерфейса или других сфер когнитивной направленности.<br /><br />Партнёры программы: Сбербанк, МТС, Институт проблем информатики и Радиология Москвы. <br/><br/> Кем работают выпускники: <ol> <li>Специалист машинного обучения</li> <li>Контролёр нейросетей</li> <li>Руководитель отдела машинного обучения</li> </ol>",

                        "Учебный план": "https://new.mospolytech.ru/upload/iblock/5dc/kompyuternaya-lingvistika-i-iskusstvennyj-intellekt-mag-2020.pdf",

                        "Карта направления": "https://new.mospolytech.ru/upload/iblock/b97/kompyuternaya-lingvistika-i-iskusstvennyj-intellekt-mag-2020-karta.pdf"

                    },

                    {

                        "Название направления": "Медицинские интеллектуальные системы",

                        "Код направления": "09.04.01",

                        "Срок обучения": "2",

                        "Бюджетных мест": "31",

                        "Подробнее":"https://fit.mospolytech.ru/programs/09.04.01.02    ",

                        "Платных мест": "10+",

                        "Описание": "Данное направление готовит специалистов для совеременных медицинских систем будщего. Обучение направлено на разработку аналитических приложений для терпапевтической сферы, программирование роботов-хирургов и создание мобильных приложений. <br /><br />В части прикладной подготовки особое внимание уделяется вопросам создания носимых гаджетов, мобильных приложений и сопутствующих веб-сервисов персональной медицины. >Партнёры программы: Сбербанк, МТС, Институт проблем информатики и Радиология Москвы. <br /><br /> Кем работают выпускники: <ol> <li>Разработчик медицинского ПО</li> <li>Специалист по машинному обучению</li> </ol>",

                        "Учебный план": "https://new.mospolytech.ru/upload/iblock/a14/medicinskie-intellektualnye-sistemy-mag-2020.pdf",

                        "Карта направления": "https://new.mospolytech.ru/upload/iblock/a9c/medicinskie-intellektualnye-sistemy-mag-2020-karta.pdf"

                    }

                ]

            },

            {

                "Информационная безопасность - 10.04.01": [

                    {

                        "Название направления": "Безопасность веб-приложений и облачных технологий",

                        "Код направления": "01.04.01",

                        "Срок обучения": "2",

                        "Бюджетных мест": "12",

                        "Платных мест": "15+",

                        "Подробнее":"https://fit.mospolytech.ru/programs/10.04.01",

                        "Описание": "Это направление готовит специалистов по безопасности крупных веб-систем и хранилищ. Как создать удобный интерфейс, продумать безопасную систему обмен  а данными, обезопасить пользователей на прикладном уровне - именно этому вы готовят студентов на этом направлении.<br/><br/> Партнёры программы: Банк России, СПМ банк, Лаборатория Касперского и бюро аналитики уязвимостей Infowatch.<br/><br/>Кем работают выпускники:<ol><li>Архитектор информационной безопасности</li><li>Специалист по тестированию и анализу уязвимостей ПО</li><li>Аудитор безопасности банковских систем</li></ol>",

                        "Учебный план": "https://new.mospolytech.ru/upload/iblock/78b/informacionnaya-bezopasnost-v-web-razrabotke-i-oblachnykh-tekhnologiyakh-mag-2020.pdf",

                        "Карта направления": " "

                    }

                ]

            },

            {

                "Управление в технических системах - 27.04.04": [

                    {

                        "Название направления": "Роботизированные беспилотные системы",

                        "Код направления": "27.04.04.02.02",

                        "Срок обучения": "2",

                        "Бюджетных мест": "10",

                        "Платных мест": "20+",

                        "Подробнее":"https://fit.mospolytech.ru/programs/27.04.04.02.02",

                        "Описание": "В программе обучения включены компетенции разработки программ для беспилотников, создания интерфейсов управления и работы с микроконтроллерами.<br/><br/>Студенты изучают весь цикл создания беспилотных и автоматизированных систем от уровня железа до пользовательских интерфейсов. Тестирование аппаратов проходят на полигоне университета.<br>Партнёрами программы являются: концерн \"Вертолёты России\", Военно-медицинская академия имени С. М. Кирова и производитель беспилотников \"Полдень 21-й Век\"<br/><br/>Профессии выпускников:<ol><li>Ведущий разработчик беспилотного ПО</li><li>Оператор автоматизированных систем</li><li>Инженер-тестировщик беспилотных устройств</li></ol>",

                        "Учебный план": " ",

                        "Карта направления": " "

                    }

                ]

            }

        ]

    },

    "Заочная": {

        "Бакалавриат": [

            {

                "Информатика и вычислительная техника - 09.03.01": [

                    {

                        "Название направления": "Программное обеспечение информационных систем",

                        "Код направления": "09.03.01",

                        "Срок обучения": "5 лет",

                        "Бюджетных мест": "25",

                        "Платных мест": "10+",

                        "Подробнее":"https://fit.mospolytech.ru/programs/09.03.01",

                        "Описание": "Студенты изучают принципы создания программного обеспечения для современных предприятий. В программу обучения входят компетенции программистов, аналитиков бизнес-процессов, проектировщиков информационных моделей.<br/><br/>Особое внимание уделяется формированию навыков проектирования, разработки и тестирования настольных, веб и мобильных приложений, автоматизирующих широкий класса бизнес-задач. В ходе обучения студенты получают сертификаты специалистов от 1С, Oracle и Cisco academy.<br/><br/> Партнёры программы: Google, Microsoft,  Cisco academy и компания 1C.<br/><br/>Кем работают выпускники:<ol><li>Специалист по информационным технологиям</li><li>Разработчик интеллектуальных систем</li><li>Руководитель разработки программного обеспечения</li></ol>  ",

                        "Учебный план": "https://new.mospolytech.ru/upload/iblock/4dc/programmnoe-obespechenie-informacionnykh-sistem-bak-2020.pdf",

                        "Карта направления": " "

                    }

                ]

            }

        ]

    }

}

1. Руководство пользователя

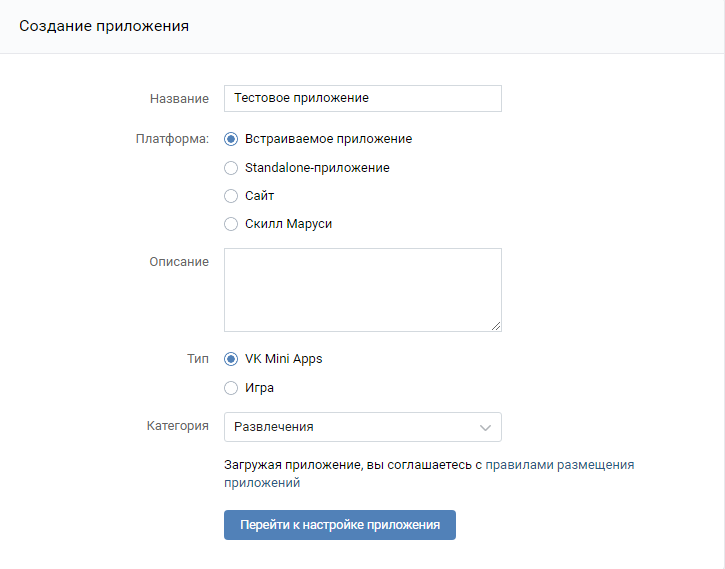
Если вы абитуриент

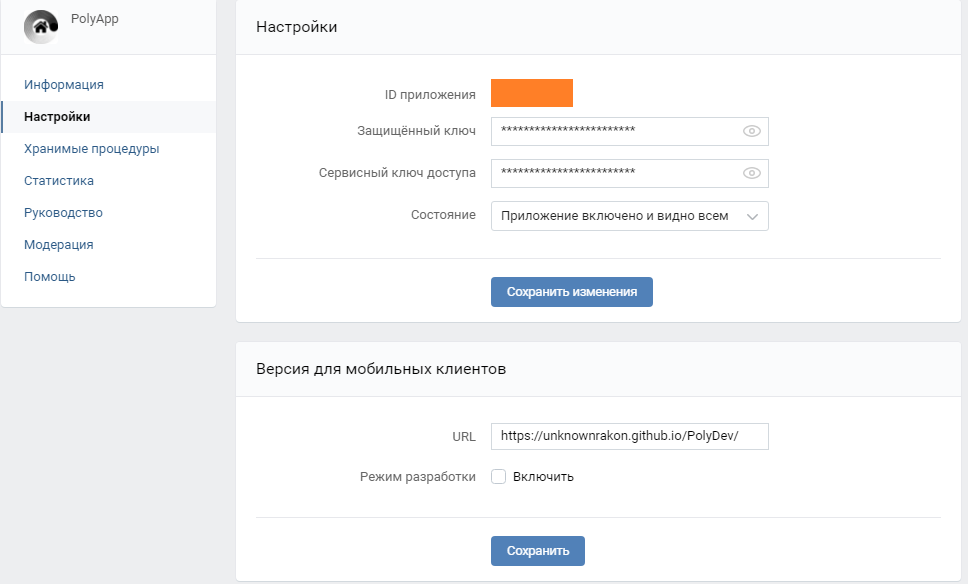
1. Нажмите по пункту «я хочу поступить» и ступень образования и форму обучения.
2. Выберите интересующие вас направления и узнайте информацию про них, нажатием по кнопке «инфо» в строке каждого направления.
3. После перехода на следующую страницу вы можете изучить какие предметы нужно сдать для поступления по данным направлениям, перейти на сайты, где можно подать документы онлайн или же перейти на страницу «Общежития».
4. На странице «Общежития» вы можете узнать у каких студентов есть право получить общежитие и изучить каждое общежитие.

Если вы студент

1. Нажмите по пункту «я уже студент»
2. Заполните о себе информацию: курс, направление, группа и общежитие. Опционально вы можете заполнить номер профбилета и студбилета.
3. На странице студента вы можете отредактировать информацию о себе или сбросить все данные и вернутся в начало приложения.
4. Внизу экрана приложения присутствует меню-бар с пактами: Частые вопросы, Календарь и страница студента.
5. В разделе «вопросы» представлены четыре категории вопросов: про учёбу, общежития, учебные корпуса и проектную деятельность
6. В разделе «календарь» у пользователей есть возможность создавать себе напоминания (не больше 5). Выберете нужную дату, текст уведомления и нажмите «создать». В указанную дату вам придёт пуш-уведомление.
7. Инструкция по установке

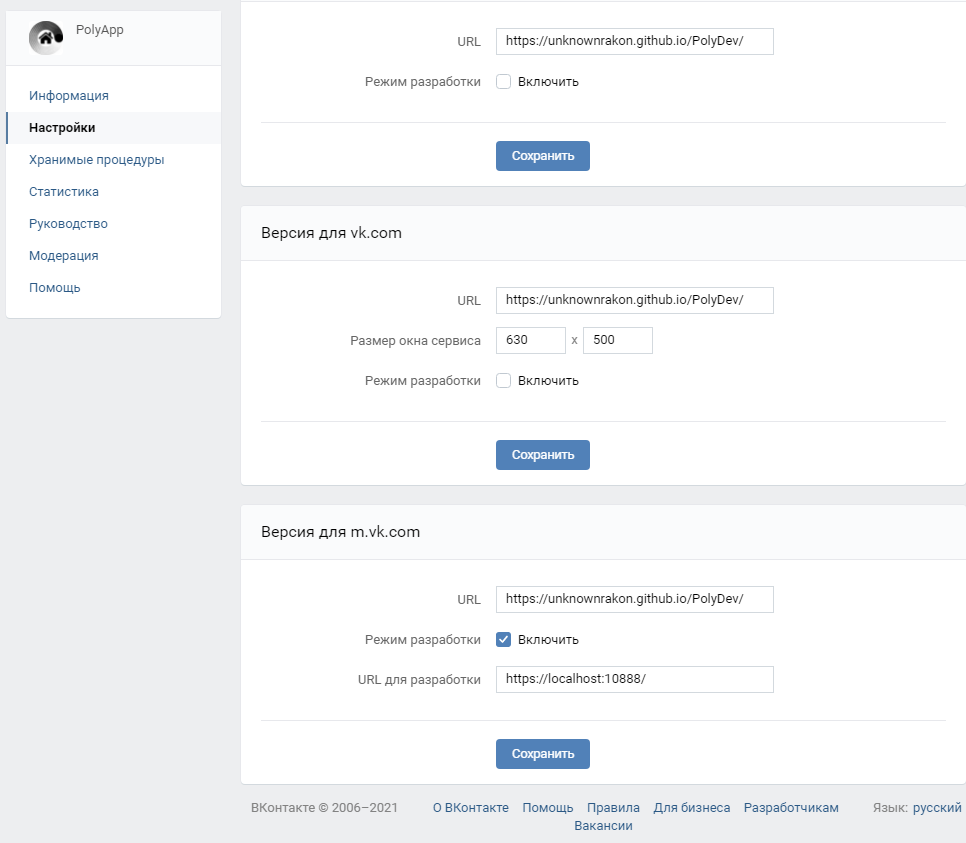
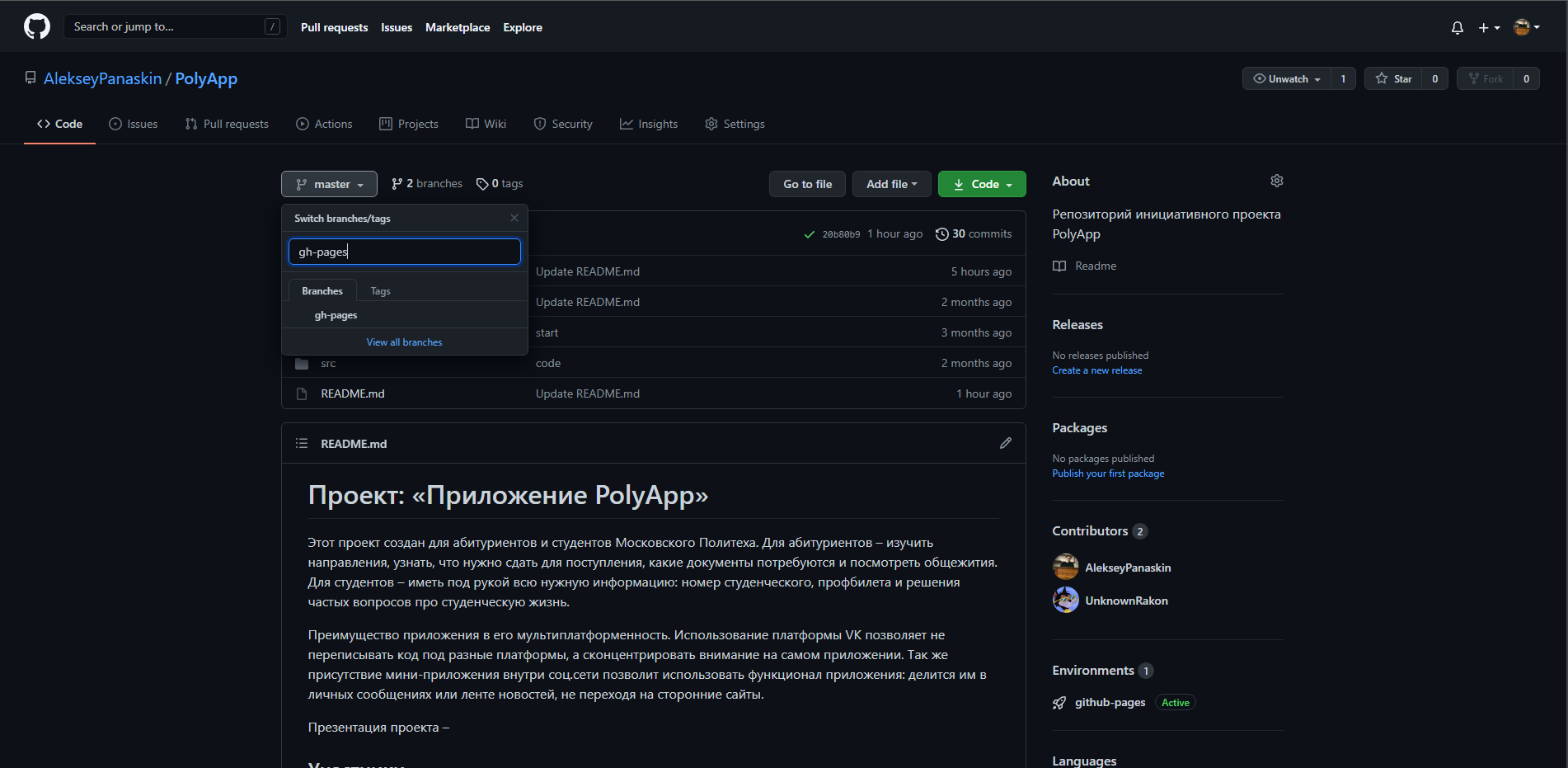
**Регистрация приложения**

[](https://github.com/AlekseyPanaskin/PolyApp/blob/master/docs/guide_1.png?raw=true)Создайте приложение на этой странице [Страница для разработчиков VK](https://vk.com/apps?act=manage)

[](https://github.com/AlekseyPanaskin/PolyApp/blob/master/docs/guide_2.png?raw=true)Скачайте [репозиторий](https://github.com/UnknownRakon/PolyDev) или создайте своё приложение на основе [быстрого шаблона](https://vk.com/dev/vk_apps_docs).

## **Локальная развёртка приложения**

Скачанный репозиторий необходимо открыть в командной строке. Заранее установите *[node-js](https://nodejs.org/en/" \o "node-js)*.

[](https://github.com/AlekseyPanaskin/PolyApp/blob/master/docs/guide_3.png?raw=true)В командной строке необходимо прописать команду npm i После установки всех пакетов, необходимо создать собственный репозиторий на [GitHub](https://github.com/). Подключите удалённый репозиторий к своему проекту и создайте ветку gh-pages. [](https://github.com/AlekseyPanaskin/PolyApp/blob/master/docs/guide_4.png?raw=true) В корневой папке проетка найдите файл package.json. Измените строку "homepage": "https://ВАШ\_НИК.github.io/НАЗВАНИЕ\_РЕПОЗИТОРИЯ/" В командной строке пропишите команду npm run deploy