**Оглавление**

[1 Django-Практика 2](#_Toc154167216)

[1.1 Общие сведения. 2](#_Toc154167217)

[1.1.1 Что такое Django 2](#_Toc154167218)

[1.1.2 Установка и настройка. 2](#_Toc154167219)

[1.1.3 Основные команды Django 2](#_Toc154167220)

[1.2 Руководство программиста 2](#_Toc154167221)

[1.2.1 Модели 2](#_Toc154167222)

[1.2.2 Представления 2](#_Toc154167223)

[1.2.3 Шаблоны 3](#_Toc154167224)

[1.2.4 Работа с формами 4](#_Toc154167225)

[1.3 Руководство пользователя 4](#_Toc154167226)

[1.3.1 Навигация 4](#_Toc154167227)

[1.3.2 Добавление позиции в базу данных. 5](#_Toc154167228)

[1.3.3 Изменение позиции меню: 5](#_Toc154167229)

[1.3.4 Удаление позиции меню: 6](#_Toc154167230)

[2 DIGLIT (Цифровая граммотность) 6](#_Toc154167231)

[2.1 Общие сведения 6](#_Toc154167232)

[2.2 Требования к проекту 7](#_Toc154167233)

[2.3 Структура заданий 7](#_Toc154167234)

[2.4 Макеты заданий 7](#_Toc154167235)

[2.5 Реализация сборки заданий и развертывание проекта 8](#_Toc154167236)

[2.5.1 Сборка заданий 8](#_Toc154167237)

[2.5.2 Развертывание проекта 8](#_Toc154167238)

[2.6 Ссылки на задания 9](#_Toc154167239)

[2.7 Верстка заданий 9](#_Toc154167240)

# Django-Практика

## Общие сведения.

### Что такое Django

**Django** - это специальный инструмент, который помогает разработчикам создавать веб-сайты и приложения. Он называется фреймворком, потому что предоставляет готовые инструменты и правила для работы с веб-сайтами.

### Установка и настройка.

Чтобы установить Джанго, нужно ввести в терминал команду:

* $ pip install Django
* После ввести в терминал:
* $ django-admin startproject
* И создаться джанго-проект.

### Основные команды Django

Важные команды, которыми желательно использовать при разработке проекта.

Создание миграций модели:

$ python manage.py makemigrations learning\_logs

Проведение миграций:

$ python manage.py migrate

Создание учтённой записи администратора:

$ python manage.py createsuperuser

## Руководство программиста

### Модели

**Модель Django** — это встроенная функция, которую Django использует для создания таблиц, их полей и различных ограничений. Проще говоря, модели Django - это база данных SQL, которую используют с Django. Модели Django упрощают задачи и организуют таблицы в модели.

Positions – это одна из моделей для базы данных позиции товаров созданная мной.

class Position(models.Model):  
 name = models.CharField('Название', max\_length=30)  
 type = models.CharField('Тип', max\_length=30)  
 gramm = models.CharField('Литры', max\_length=10)  
 sostav = models.TextField('Описание')  
 price = models.IntegerField('Цена')  
  
 def \_\_str\_\_(self):  
 return self.name  
 class Meta:  
 verbose\_name="Позиция"  
 verbose\_name\_plural="Позиции товаров"

### Представления

В Django представления (views) — это код, который управляет тем, что отображается в браузере после запроса пользователя на сервер.

Для создания представления в Django, необходимо определить функцию, которая будет получать запрос и возвращать объект ответа HTTP или вызвать метод, которые Django предоставляет для создания ответа.

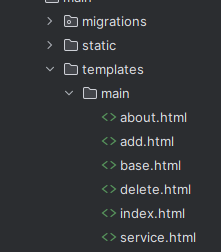
Код файла views.py:

def index(request):  
 positions\_all = Position.objects.all()  
 positions = PostFilter(request.GET, queryset=positions\_all)  
 return render(request, 'main/index.html', {'title': 'Главная страница сайта', 'positions': positions})  
  
  
def about(request):  
 return render(request, 'main/about.html')  
  
  
def service(request):  
 return render(request, 'main/service.html')  
  
  
def add(request):  
 error = ''  
 if request.method == 'POST':  
 form = PosForm(request.POST)  
 if form.is\_valid():  
 form.save()  
 return redirect('home')  
 else:  
 error = 'Форма заполнена неправильно или не до конца'  
 form = PosForm()  
 context = {  
 'form': form  
 }  
 return render(request, 'main/add.html', context)  
  
  
class Delete(DeleteView):  
 model = Position  
 success\_url = '/'  
 template\_name = 'main/delete.html'  
  
  
class Update(UpdateView):  
 model = Position  
 template\_name = 'main/add.html'  
 form\_class = PosForm  
 success\_url = '/'

### Шаблоны

**Шаблоны** **Django** — **это** файлы, которые определяют, как должен выглядеть HTML-код страниц веб-приложения.

Шаблоны хранятся в папке templates находящейся в директории main/templates/main.



### Работа с формами

В HTML форма представляет собой набор элементов внутри <form>...</form>, которые позволяют посетителю выполнять такие действия, как ввод текста, выбор параметров, манипулирование объектами или элементами управления и так далее, а затем отправлять эту информацию обратно на сервер.

GET и POST - единственные HTTP-методы, которые можно использовать при работе с формами.

Форма входа в систему Django возвращается с использованием POST метода, в котором браузер объединяет данные формы, кодирует их для передачи, отправляет на сервер, а затем получает обратно свой ответ.

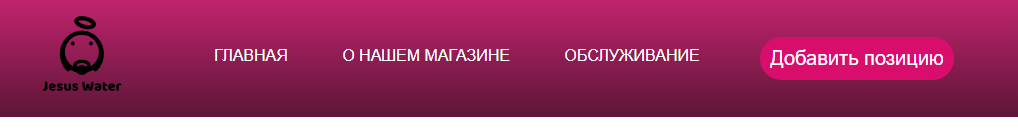
Код в файле forms.py:

class PosForm(ModelForm):  
 class Meta:  
 model = Position  
 fields = ["name", "type", "gramm", "sostav", "price"]  
 widgets = {  
 "name": TextInput(attrs={'placeholder': 'Название','class': 'form-input'}),  
 "type": TextInput(attrs={'placeholder': 'Тип','class': 'form-input'}),  
 "gramm": TextInput(attrs={'placeholder': 'Литры','class': 'form-input'}),  
 "sostav": Textarea(attrs={'placeholder': 'Описание','class': 'form-input'}),  
 "price": TextInput(attrs={'placeholder': 'Цена в руб.','class': 'form-input'}),

## Руководство пользователя

### Навигация

Главное меню сверху сайта это навигация по сайту.



«Главная» - начальная и главная страница.

«О нашем магазине» - переносит на страницу с информацией о магазине.

«Обслуживание» - переносит на страницу с информацией об обслуживаня.

### Добавление позиции в базу данных.

|  |  |
| --- | --- |
| В главном меню выберите пункт «Добавить позицию» |  |
| Заполните поля формы. |  |
| Нажать «Добавить» |  |
| Вас переадресует на Главную страницу, где будет отображена новая позиция. |  |

### Изменение позиции меню:

|  |  |
| --- | --- |
| Перейдите на Главную страницу |  |
| В строке с позицией нажмите кнопку «Изменить» |  |
| Измените нужные поля. |  |
| Нажмите кнопку «Добавить» |  |

### Удаление позиции меню:

|  |  |
| --- | --- |
| Перейдите на Главную страницу |  |
| В строке с позицией нажмите кнопку «Удалить» |  |
| Подтвердите удаление |  |

# DIGLIT (Цифровая граммотность)

## Общие сведения

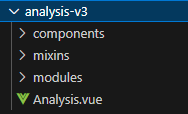
Цифровая грамотность сложный для измерения конструкт, поскольку включает в себя не только способность использовать цифровые технологии для работы с информацией, но и информационную безопасность, этику, способность работать с нелинейным и динамическим материалом, фильтровать большой объем информации и т.д. Кроме этого, цифровая грамотность связана с критическим мышлением, коммуникацией, сотрудничеством, техническими навыками работы с определенными инструментами (например, браузером, почтовым клиентом и проч.), что накладывает дополнительные сложности для её измерения. Поэтому цифровую грамотность невозможно измерить традиционными тестами по компьютерной грамотности, которые измеряют прежде всего технические навыки работы с компьютером.

## Требования к проекту

Тех. требования: поддержка Google Chrome последних версий (>= 120.0.6099.62); Разрешение экрана 1024x768; поддержка ввода с клавиатуры/мыши.

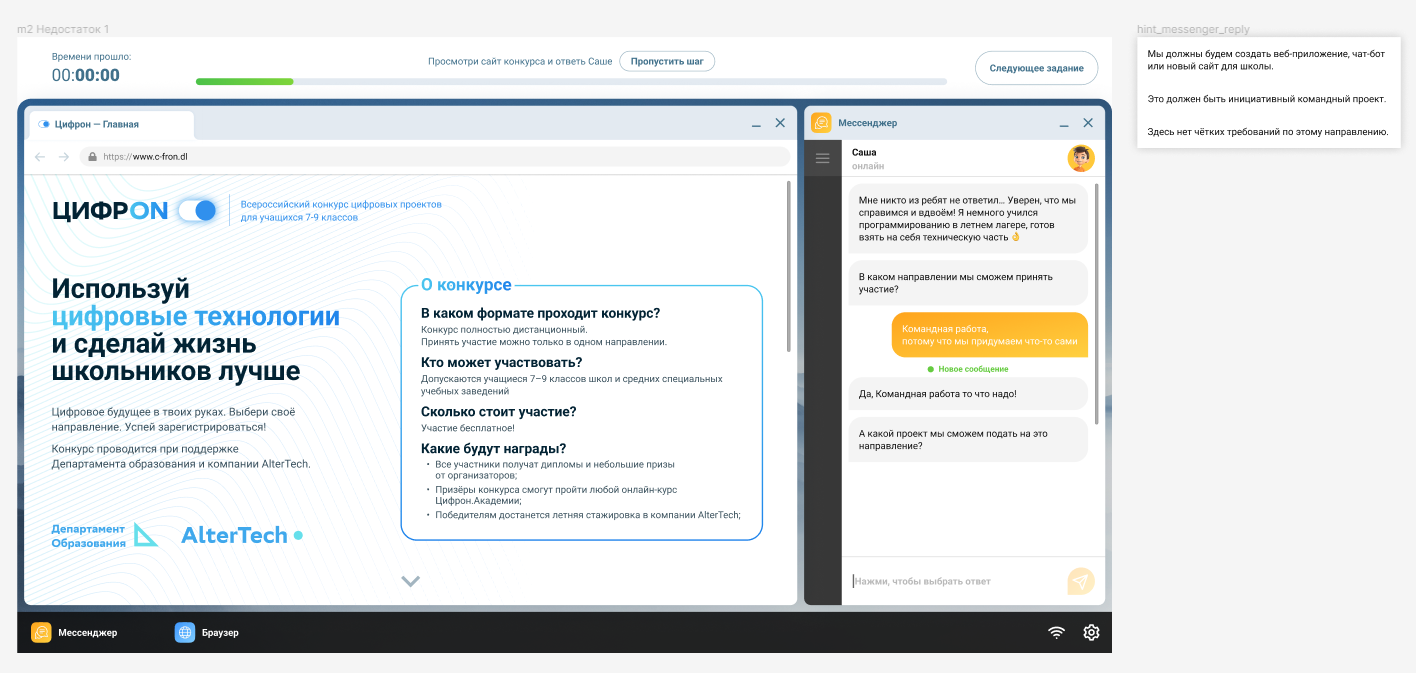
Задания представляют собой симуляторы с логикой распространенных приложений: браузер, мессенджер, графический редактор, блокнот и тп. В заданиях должен быть предусмотрен сбор индикаторов для их последующей отправки в систему скоринга на базе байесовской сети. Индикатор – элемент ассоциативного массива (ключ/значение), где ключ – уникальный идентификатор узла байесовского графа строчного типа, и значение – дихотомическое (0 или 1) или политомическое (0, 1, 2, …) представление числового типа. Задания должны состоять из нескольких модулей (отдельных изолированных частей/экранов), в соответствии со сценарием.

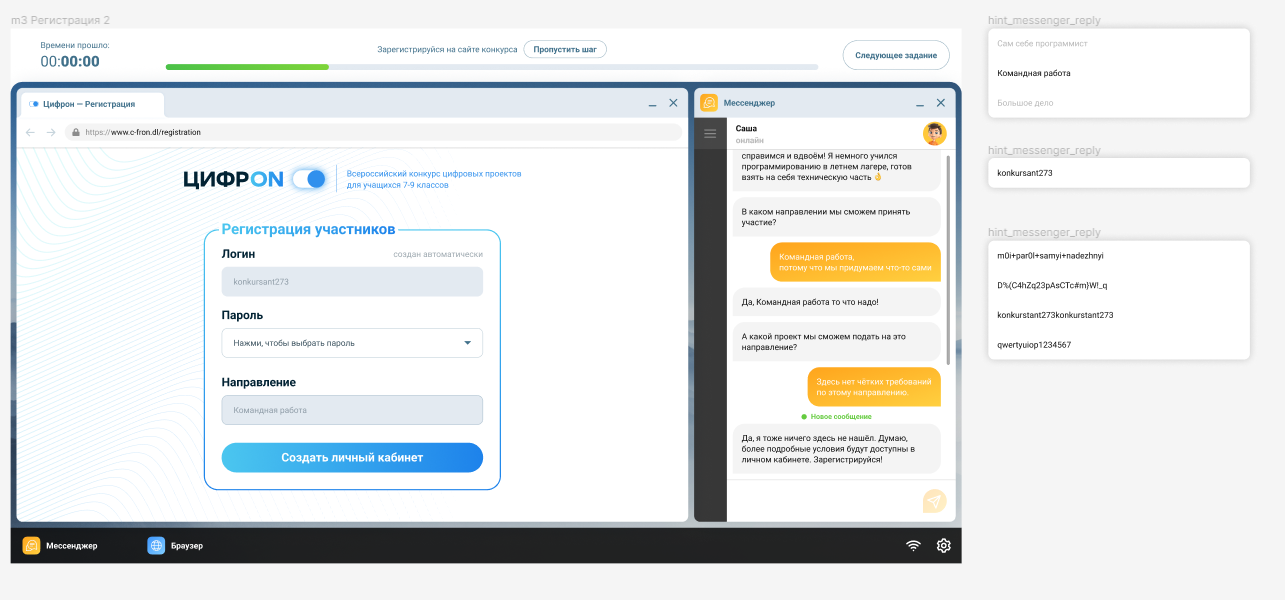
## Структура заданий



* \* - корневая папка задания в библиотеке заданий;
* components - папка с локальными компонентами, используемые в данном задании (Кнопки, Модальные окна, Конструкторы и прочее);
* mixins - папка содержащая Vue-миксины, необходимые в данном задании;
* modules - модули задания;
* \*.vue - файл-компонент, в котором происходит регистрация задания в системе (инициализация).

## Макеты заданий





## Реализация сборки заданий и развертывание проекта

### Сборка заданий

Для сборки заданий используется бандлер Webpack со следующими основными loader’ами:

* vue-loader - поддержка Vue Single-File Companents;
* babel-loader - транспиляция javascript, а также инъекции полифиллов;
* sass-loader/postcss-loader - препроцессоры для SASS и SCSS файлов.

Команды проекта (npm-скрипты):

* npm run dev - сборка проекта в режиме отладки для активной разработки и поддержки кода;
* npm run build - продакшн-сборка для master-сервера.

### Развертывание проекта

**## \*\*Инициализация\*\* ##**

- Склонировать проект в папку для запуска веб-сервером (`/var/www/html...`)

- После клонирования репозитория выполнить:

```

git submodule init

git submodule update --remote --checkout --recursive

```

**###** *\*backend\** **###**

- Создать новую БД

- Настроить подключение к БД:

    - Создать `www/backend/config/db.php` используя `www/backend/config/db.php.sample`

- Применить миграции:

```shell

cd www/backend/

php yii migrate

```

**###** *\*frontend\** **###**

- Выполнить команду:

```shell

cd www/frontend/

npm i

```

## Ссылки на задания

<https://dl.hse.ru/demo/2103dc87c850cf146fb34f881633504c>

<https://dl.hse.ru/demo/89f96388685ac3f93fbea7dafe43ab44>

<https://dl.hse.ru/demo/637ff66d7855c802c9191b7c39a11979>

<https://dl.hse.ru/demo/16da0f31a7c1d9739d6f526dbed195e3>

## Верстка заданий

Пример одной из работ:

<template>

    <Module id="A0" class="A0" :state="$data" :indicators="calcIndicators" ref="module">

        <Page style="background-color:#222222">

            <TaskIntroDL8V3

                :task="7"

                :hide="!modules.A0.intro"

                @start="

                    modules.A0.intro = false;

                    openApp('moduleplus');

                    if( !modules.A0.startedAt )

                        modules.A0.startedAt = Date.now();

                " />

            <ConfirmFinish

                :hide="!modules.A0.confirmFinish"

                :timerData="timerData"

                @accept="

                    modules.A0.taskSkipped = true;

                    nextTask($refs.module);

                "

                @cancel="

                    modules.A0.confirmFinish = false;

                    modules.A0.skipTaskWarned = true;

                " />

            <Desktop3

                class="desktop"

                :apps="apps"

                :shortcuts="shortcuts"

                :sleep-mode="modules.A0.intro || modules.A0.confirmFinish"

                :timerData.sync="timerData"

                :startedAt="modules.A0.startedAt"

                :skipStepWarned.sync="modules.A0.skipStepWarned"

                @skip-warned="modules.A0.skipStepWarned = true"

                @skip="skipSteps()"

                @next="

                    modules.A0.nextBtnClickCount++;

                    toNextTask();

                "

            >

                <template #moduleplus="{ app }">

                    <Moduleplus

                        ref="moduleplus"

                        :app="app"

                        :app-data="appData.moduleplus"

                        :commands="commands"

                        :sidebar="!modules.A0.taskDone"

                        :editable="!modules.A0.taskDone"

                        arrow-mode-always

                        bar-background="#E3EAF1"

                        can-minimize>

                        <ScrollBox

                            class="task-sidebar"

                            :bar-width="4"

                            :thumb-padding="0"

                            style="width:300px">

                            <div class="d-flex flex-column w-100 h-100" style="gap:20px">

                                <div class="flex-fill">

                                    <p>Составь блок-схему по описанию ниже:</p>

                                    <ul>

                                        <li>Сначала  выбирается один аспект работы школьной библиотеки (например, режим работы, наличие книг, мероприятия)  с помощью  функции случайного выбора.</li>

                                        <li>После чего бот  должен выводить приветстветственное обращение к пользователю с просьбой оценить выбранный аспект работы школьной библиотеки.</li>

                                        <li>Затем идет поиск вопроса о работе школьной библиотеки, касающегося выбранного аспекта (к примеру: на сколько ты удовлетворен режимом работы школьной библиотеки), который нужно оценить.</li>

                                        <li>Пользователь выставляет оценку от 1 до 5. Оценка записывается на сервер.</li>

                                        <li>Если поставленная оценка ниже 3, бот должен выводить текст, в котором просит пояснить свою оценку – оставить отзыв о работе школьной библиотеки. Пользователь вводит текст, который также необходимо записать на сервер.</li>

                                        <li>Помимо текста в отзыве пользователь может прикрепить файл. Следовательно, необходимо спросить пользователя о желании добавить файл</li>

                                        <li>Если выбранный файл картинка, не забудь записать отзыв на сервер!</li>

                                        <li>Многие любят присылать видео- и аудиофайлы. Но эта информация будет довольно долго обрабатываться, поэтому, если загруженный файл НЕ картинка, нужно вывести текст “Не удается выбрать файл данного формата”. Проверка формата файла повторяется/осуществляется до тех пор, пока выбранный файл не будет картинкой.</li>

                                        <li>После этого бот прощается с пользователем.</li>

                                    </ul>

                                    <p class="m-0">Когда закончишь работу, <b>нажми кнопку «Готово»</b>.</p>

                                </div>

                                <button

                                    @click="

                                        if( !!appData.moduleplus.blockDiagram.length )

                                            nextStep();

                                    "

                                    class="task-btn w-100"

                                    :class="{ 'disabled': !appData.moduleplus.blockDiagram.length || modules.A0.taskDone }"

                                    type="button">

                                    <span>Готово</span>

                                </button>

                            </div>

                        </ScrollBox>

                    </Moduleplus>

                </template>

            </Desktop3>

        </Page>

    </Module>

</template>

<script>

    import \_ from 'lodash'

    import test from '~/mixins/test'

    import apps from '~/mixins/apps'

    import pausesCapture from '~/mixins/pausesCapture'

    import clicksCapture from '~/mixins/clicksCapture'

    import { app } from '~/components/global/Desktop3'

    import { newBlock } from '~/components/apps/Moduleplus'

    const { v4: uuid } = require('uuid');

    const NA = 'NA';

    const MISSING = 'missing';

    const CMD\_BEGIN = 'Начало';

    const CMD\_END = 'Конец';

    const CMD\_HI = 'Привет! Оцени, пожалуйста, работу школьной библиотеки';

    const CMD\_SEARCH\_PHOTO = 'Вопрос о работе школьной библиотеки найден в базе';

    const CMD\_OUTPUT\_PHOTO = 'Вопрос выведен из базы';

    const CMD\_RATING = 'Оценка от 1 до 5';

    const CMD\_SAVE\_RATING = 'Запись оценки на сервер';

    const CMD\_RATING\_LESS\_3 = 'Поставленная оценка ниже 3х';

    const CMD\_REVIEW = 'Оставь отзыв работе школьной библиотеки';

    const CMD\_INPUT\_REVIEW = 'Ввод отзыва';

    const CMD\_ATTACH\_FILE = 'Хочешь прикрепить файл?';

    const CMD\_ADD\_FILE = 'Добавить файл';

    const CMD\_SELECT\_FILE = 'Выбор файла';

    const CMD\_FILE\_IS\_IMG = 'Выбранный файл картинка';

    const CMD\_FILE\_ERROR = 'Не удается выбрать файл данного формата';

    const CMD\_FILE\_LOADING = 'Загрузка файла';

    const CMD\_SAVE\_REVIEW = 'Запись отзыва на сервер';

    const CMD\_THANKS = 'Спасибо за оценку работы школьной библиотеки!';

    const CMD\_RANDOM = 'Вопрос о работе школьной библиотеки выбран случайным образом из списка имеющихся';

    export default {

        mixins: [test, apps, pausesCapture('A0'), clicksCapture('A0')],

        created()

        {

            if( !('A0' in this.modules) )

                this.modules = {

                    ...this.modules,

                    A0: {

                        time: Date.now(),

                        openedAt: Date.now(),

                        startedAt: 0,

                        pauses: [],

                        clicks: [],

                        stepsEndedAt: [0],

                        intro: true,

                        confirmFinish: false,

                        taskDone: false,

                        taskSkipped: false,

                        skipStepWarned: false,

                        skipTaskWarned: false,

                        nextBtnClickCount: 0,

                        timerData: {

                            steps: [

                                {

                                    name: 'Построй блок-схему согласно присланному описанию',

                                    skipped: false,

                                    complete: false,

                                },

                                {

                                    name: 'Построй блок-схему согласно присланному описанию',

                                    skipped: false,

                                    complete: false,

                                },

                            ],

                            currentStep: 1,

                            skipOpen: false,

                        },

                        apps: {

                            moduleplus: this.initApp('moduleplus'),

                        },

                        appData: {

                            moduleplus: {

                                blockDiagram: [],

                            },

                        },

                    }

                }

        },

        computed: {

            \_: () => \_,

            NA: () => NA,

            MISSING: () => MISSING,

            newBlock: () => newBlock,

            apps()

            {

                return Object.values(

                    this.modules.A0.apps

                );

            },

            appData()

            {

                return this.modules.A0.appData;

            },

            timerData()

            {

                return this.modules.A0.timerData;

            },

            shortcuts()

            {

                return [

                    {

                        name: 'Элемент+',

                        icon: require('~/assets/icons/apps3/icon-moduleplus.svg'),

                        action: focus => {

                            focus(this.openApp('moduleplus'));

                        }

                    },

                ]

            },

            commands()

            {

                const cmds = [

                    CMD\_BEGIN,

                    CMD\_END,

                    CMD\_HI,

                    CMD\_SEARCH\_PHOTO,

                    CMD\_OUTPUT\_PHOTO,

                    CMD\_RATING,

                    CMD\_SAVE\_RATING,

                    CMD\_RATING\_LESS\_3,

                    CMD\_REVIEW,

                    CMD\_INPUT\_REVIEW,

                    CMD\_ATTACH\_FILE,

                    CMD\_ADD\_FILE,

                    CMD\_SELECT\_FILE,

                    CMD\_FILE\_IS\_IMG,

                    CMD\_FILE\_ERROR,

                    CMD\_FILE\_LOADING,

                    CMD\_SAVE\_REVIEW,

                    CMD\_THANKS,

                    CMD\_RANDOM,

                ];

                return cmds

                    .filter(cmd => !(

                        this.appData.moduleplus.blockDiagram

                            .find(block => block.text == cmd)

                    ));

            },

            arrows()

            {

                const { blockDiagram } = this.appData.moduleplus;

                const arrows = [];

                for(let block of blockDiagram)

                    for(let slot in block.out)

                        for(let id of block.out[slot])

                            arrows.push({

                                from: block,

                                slot,

                                to: blockDiagram

                                    .find(b => b.id == id),

                            });

                return arrows;

            },

            time()

            {

                return this.modules.A0.time - this.modules.A0.startedAt;

            },

        },

        methods: {

            initApp(appName)

            {

                let A0 = this.modules.A0;

                switch(appName){

                    case 'moduleplus':

                        return app({

                            slot: 'moduleplus',

                            name: 'Элемент+',

                            icon: require('~/assets/icons/apps3/icon-moduleplus.svg'),

                            barHeight: '50px',

                            iconWidth: '30px',

                            iconHeight: '30px',

                            window: {

                                position: {

                                    x: '10px',

                                    y: '10px',

                                },

                                size: {

                                    w: 'calc(100% - 20px)',

                                    h: 'calc(100% - 20px)',

                                },

                            },

                        });

                    break;

                }

            },

            blur()

            {

                for(let { window } of this.apps)

                    window.focused = false;

            },

            openApp(appName)

            {

                this.blur();

                if( !(appName in this.modules.A0.apps) )

                    this.modules.A0.apps = {

                        ...this.modules.A0.apps,

                        [appName]: { ...this.initApp(appName) },

                    };

                this.modules.A0.apps[appName].window = {

                    ...this.modules.A0.apps[appName].window,

                    minimized: false,

                    focused: true,

                    zIndex: this.apps.length,

                };

                return this.modules.A0.apps[appName];

            },

            nextStep()

            {

                this.modules.A0.stepsEndedAt[this.timerData.currentStep - 1] = Date.now();

                this.timerData.currentStep++;

                this.modules.A0.taskDone = true;

                this.nextTask(this.$refs.module);

            },

            // skips

            skipSteps() {

                this.timerData.steps[this.timerData.currentStep - 1].skipped = true;

                this.nextStep();

            },

            toNextTask()

            {

                if( this.modules.A0.taskDone )

                    return this.nextTask(this.$refs.module);

                this.modules.A0.confirmFinish = true;

            },

            calcIndicators(indicators)

            {

                let A0 = this.modules.A0;

                // m0

                indicators['t08c\_m00\_LT\_8'] = (() => {

                    return Math.max(0, (

                        Math.round(

                            (A0.startedAt - A0.openedAt) / 1000

                        )

                    ));

                })();

                // m1

                indicators['t08c\_m01\_LT\_8'] = (() => {

                    return Math.max(0, (

                        Math.round(

                            (A0.stepsEndedAt[0] - A0.startedAt) / 1000

                        )

                    ));

                })();

                const relation = (a, b, fromSlot = null) => (

                    this.arrows.find(({ from, to, slot }) => (

                        from.text == a

                        &&

                        to.text == b

                        &&

                        (

                            fromSlot === null

                            ||

                            slot == fromSlot

                        )

                    ))

                )

                indicators['t08c\_m01\_CLF3-1\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    if( relation(CMD\_BEGIN, CMD\_RANDOM) )

                        return 1;

                    return 0;

                })();

                indicators['t08c\_m01\_CLF3-2\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    if( relation(CMD\_RANDOM, CMD\_HI) )

                        return 1;

                    return 0;

                })();

                indicators['t08c\_m01\_CLF3-3\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    if( relation(CMD\_HI, CMD\_SEARCH\_PHOTO) )

                        return 1;

                    return 0;

                })();

                indicators['t08c\_m01\_CLF3-4\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    if( relation(CMD\_SEARCH\_PHOTO, CMD\_OUTPUT\_PHOTO) )

                        return 1;

                    return 0;

                })();

                indicators['t08c\_m01\_CLF3-5\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    if( relation(CMD\_OUTPUT\_PHOTO, CMD\_RATING) )

                        return 1;

                    return 0;

                })();

                indicators['t08c\_m01\_CLF3-6\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    if( relation(CMD\_RATING, CMD\_SAVE\_RATING) )

                        return 1;

                    return 0;

                })();

                indicators['t08c\_m01\_CLF3-7\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    if( relation(CMD\_SAVE\_RATING, CMD\_RATING\_LESS\_3) )

                        return 1;

                    return 0;

                })();

                indicators['t08c\_m01\_CLF3-8\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    if( relation(CMD\_RATING\_LESS\_3, CMD\_REVIEW, 'left') )

                        return 1;

                    return 0;

                })();

                indicators['t08c\_m01\_CLF3-9\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    if( relation(CMD\_RATING\_LESS\_3, CMD\_THANKS, 'right') )

                        return 1;

                    return 0;

                })();

                indicators['t08c\_m01\_CLF3-10\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    if( relation(CMD\_THANKS, CMD\_END) )

                        return 1;

                    return 0;

                })();

                indicators['t08c\_m01\_CLF3-11\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    if( relation(CMD\_REVIEW, CMD\_INPUT\_REVIEW) )

                        return 1;

                    return 0;

                })();

                indicators['t08c\_m01\_CLF3-12\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    if( relation(CMD\_INPUT\_REVIEW, CMD\_ATTACH\_FILE) )

                        return 1;

                    return 0;

                })();

                indicators['t08c\_m01\_CLF3-13\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    if( relation(CMD\_ATTACH\_FILE, CMD\_ADD\_FILE) )

                        return 1;

                    return 0;

                })();

                indicators['t08c\_m01\_CLF3-14\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    if( relation(CMD\_ADD\_FILE, CMD\_SELECT\_FILE, 'left') )

                        return 1;

                    return 0;

                })();

                indicators['t08c\_m01\_CLF3-15\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    if( relation(CMD\_ADD\_FILE, CMD\_SAVE\_REVIEW, 'right') )

                        return 1;

                    return 0;

                })();

                indicators['t08c\_m01\_CLF3-16\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    if( relation(CMD\_SELECT\_FILE, CMD\_FILE\_IS\_IMG) )

                        return 1;

                    return 0;

                })();

                indicators['t08c\_m01\_CLF3-17\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    if( relation(CMD\_FILE\_IS\_IMG, CMD\_FILE\_LOADING, 'left') )

                        return 1;

                    return 0;

                })();

                indicators['t08c\_m01\_CLF3-18\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    if( relation(CMD\_FILE\_IS\_IMG, CMD\_FILE\_ERROR, 'right') )

                        return 1;

                    return 0;

                })();

                indicators['t08c\_m01\_CLF3-19\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    if( relation(CMD\_FILE\_LOADING, CMD\_SAVE\_REVIEW) )

                        return 1;

                    return 0;

                })();

                indicators['t08c\_m01\_CLF3-20\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    if( relation(CMD\_SAVE\_REVIEW, CMD\_THANKS) )

                        return 1;

                    return 0;

                })();

                indicators['t08c\_m01\_CLF3-21\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    if( relation(CMD\_THANKS, CMD\_END) )

                        return 1;

                    return 0;

                })();

                indicators['t08c\_m01\_L0\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    const { blockDiagram } = this.appData.moduleplus;

                    const commands = [

                        CMD\_BEGIN,

                        CMD\_RANDOM,

                        CMD\_HI,

                        CMD\_SEARCH\_PHOTO,

                        CMD\_OUTPUT\_PHOTO,

                        CMD\_RATING,

                        CMD\_SAVE\_RATING,

                        CMD\_RATING\_LESS\_3,

                        CMD\_REVIEW,

                        CMD\_THANKS,

                        CMD\_END,

                        CMD\_INPUT\_REVIEW,

                        CMD\_ATTACH\_FILE,

                        CMD\_ADD\_FILE,

                        CMD\_SELECT\_FILE,

                        CMD\_SAVE\_REVIEW,

                        CMD\_SELECT\_FILE,

                        CMD\_FILE\_LOADING,

                        CMD\_FILE\_IS\_IMG,

                        CMD\_FILE\_ERROR,

                        CMD\_END,

                    ];

                    const types = {

                        limiter: 0,

                        function: 1,

                        inout: 2,

                        if: 3,

                    };

                    return commands

                        .map(text => (

                            types?.[

                                blockDiagram

                                    .find(block => block.text == text)

                                    ?.type

                            ] ?? '-'

                        ))

                        .join(',');

                })();

                indicators['t08c\_m01\_CLF4\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    return Number(

                        !!this.appData.moduleplus.blockDiagram.find(block => (

                            block.type == 'if'

                            &&

                            block.text == CMD\_RATING\_LESS\_3

                        ))

                    );

                })();

                indicators['t08c\_m01\_CLF5\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return MISSING;

                    if( !this.modules.A0.taskDone )

                        return NA;

                    return Number(

                        !!this.appData.moduleplus.blockDiagram.find(block => (

                            block.type == 'if'

                            &&

                            block.text == CMD\_FILE\_IS\_IMG

                        ))

                    );

                })();

                indicators['t08c\_m01\_LMM\_8'] = (() => {

                    if( this.timerData.steps[0].skipped )

                        return 1;

                    return 0;

                })();

                indicators['t08c\_m01\_LXA\_8'] = (() => {

                    return A0.nextBtnClickCount;

                })();

                indicators['t08c\_m01\_LXX\_8'] = (() => {

                    if( this.modules.A0.taskSkipped )

                        return 1;

                    return 0;

                })();

                //pauses

                const now = Date.now();

                const moduleTimeRanges = [

                    [ A0.openedAt, A0.startedAt ],

                    [ A0.startedAt, A0.stepsEndedAt[0] ],

                ];

                for(let [ i, range ] of moduleTimeRanges.entries()){

                    const [ start, end ] = range;

                    const pauses = A0.pauses

                        .filter(([ from, to ]) => (

                            from > (start || now)

                            &&

                            from <= (end || now)

                            &&

                            to !== null

                        ));

                    const clicks = A0.clicks

                        .filter(t => (

                            t > (start || now)

                            &&

                            t <= (end || now)

                        ));

                    const prefix = `t08c\_m${ i.toString().padStart(2, 0) }`;

                    indicators[`${ prefix }\_P\_PD\_8`] = (() => {

                        return (

                            pauses

                                .map(([ from, to ]) => Math.max(0, (

                                    Math.round(

                                        (to - from) / 1000

                                    )

                                )))

                                .join(',')

                            ||

                            NA

                        );

                    })();

                    indicators[`${ prefix }\_P\_PN\_8`] = (() => {

                        return pauses.length;

                    })();

                    indicators[`${ prefix }\_T\_FC\_8`] = (() => {

                        return Math.max(0, (

                            Math.round(

                                ((clicks[0] ?? 0) - (start || now)) / 1000

                            )

                        ));

                    })();

                    indicators[`${ prefix }\_C\_GN\_8`] = (() => {

                        return clicks.length;

                    })();

                }

                return indicators;

            }

        },

        components: {

            Intro: () => import('../components/Intro'),

            ConfirmFinish: () => import('../components/ConfirmFinish'),

        },

    }

</script>

<style lang="scss" scoped>

.A0{

    font-family: 'Roboto';

    ul{

        margin: 0;

        padding-left: 25px;

        li{

            margin-bottom: 0.7rem;

        }

    }

    .task-sidebar{

        position: absolute;

        padding: 20px 20px 0;

        top: 10px;

        right: 10px;

        bottom: 10px;

        width: 400px;

        background: #FFFFFF;

        border-radius: 10px;

        font-weight: 400;

        font-size: 14px;

        line-height: 130%;

        color: #101010;

    }

    .task-btn{

        margin-bottom: 20px;

        padding: 14px 29px;

        background: linear-gradient(131.7deg, #5F8BFF -40.88%, #F032BB 138.35%);

        border: 1px solid transparent;

        border-radius: 5px;

        font-weight: 700;

        font-size: 16px;

        line-height: 19px;

        text-align: center;

        color: #FFFFFF;

        &.outline{

            background: transparent;

            border: 1px solid #966AE5;

            color: #966AE5;

        }

        &.disabled{

            opacity: 0.3;

            pointer-events: none;

        }

    }

}

</style>