

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ITMO University**

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
GRADUATION THESIS**

**Разработка программного обеспечения с элементами анализа данных для
образовательного учреждения.**

Обучающийся / Student Кириллов Николай Александрович

Факультет/институт/кластер/ Faculty/Institute/Cluster факультет

инфокоммуникационных технологий

Группа/Group K34422

Направление подготовки/ Subject area 45.03.04 Интеллектуальные системы в гуманитарной сфере

Образовательная программа / Educational program Интеллектуальные системы в гуманитарной сфере 2018


Язык реализации ОП / Language of the educational program Русский

Статус ОП / Status of educational program

Квалификация/ Degree level Бакалавр

Руководитель ВКР/ Thesis supervisor Гусарова Наталия Федоровна, старший научный сотрудник, кандидат технических наук, Университет ИТМО, факультет инфокоммуникационных технологий, доцент (квалификационная категория "ординарный доцент")

Обучающийся/Student


Документ подписан	
Кириллов Николай Александрович	
23.05.2022	

(эл. подпись/ signature)

Кириллов
Николай
Александрович

(Фамилия И.О./ name
and surname)

Руководитель ВКР/
Thesis supervisor

Документ подписан	
Гусарова Наталия Федоровна	
23.05.2022	

(эл. подпись/ signature)

Гусарова
Наталия
Федоровна

(Фамилия И.О./ name
and surname)

Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ITMO University

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ /
OBJECTIVES FOR A GRADUATION THESIS

Обучающийся / Student Кириллов Николай Александрович
Факультет/институт/кластер/ Faculty/Institute/Cluster факультет
инфокоммуникационных технологий
Группа/Group K34422
Направление подготовки/ Subject area 45.03.04 Интеллектуальные системы в
гуманитарной сфере
Образовательная программа / Educational program Интеллектуальные системы в
гуманитарной сфере 2018
Язык реализации ОП / Language of the educational program Русский
Статус ОП / Status of educational program
Квалификация/ Degree level Бакалавр
Тема ВКР/ Thesis topic Разработка программного обеспечения с элементами анализа
данных для образовательного учреждения.
Руководитель ВКР/ Thesis supervisor Гусарова Наталия Федоровна, старший научный
сотрудник, кандидат технических наук, Университет ИТМО, факультет
инфокоммуникационных технологий, доцент (квалификационная категория "ординарный
доцент")

Основные вопросы, подлежащие разработке / Key issues to be analyzed

Разработка модулей системы для учреждения «Техническая школа» ГУП «Петербургский метрополитен», таких как панель для анализа данных студентов и модуль экспорта данных из системы в полноценный документ.

Дата выдачи задания / Assignment issued on: 14.02.2022

Срок представления готовой ВКР / Deadline for final edition of the thesis 28.05.2022

Характеристика темы ВКР / Description of thesis subject (topic)

Тема в области прикладных исследований / Subject of applied research: да / yes

СОГЛАСОВАНО / AGREED:

Руководитель ВКР/
Thesis supervisor

Документ подписан	
Гусарова Наталия Федоровна	

Гусарова

22.05.2022

Наталья
Федоровна

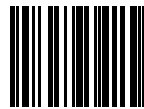
(эл. подпись)

Задание принял к
исполнению/ Objectives
assumed BY

Документ
подписан

Кириллов
Николай
Александрович

22.05.2022



Кириллов
Николай
Александрович

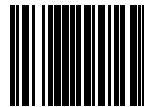
(эл. подпись)

Руководитель ОП/ Head
of educational program

Документ
подписан

Хлопотов
Максим
Валерьевич

27.05.2022



Хлопотов
Максим
Валерьевич

(эл. подпись)

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО
ITMO University**

**АННОТАЦИЯ
ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ
SUMMARY OF A GRADUATION THESIS**

Обучающийся / Student Кириллов Николай Александрович

Факультет/институт/кластер/ Faculty/Institute/Cluster факультет
инфокоммуникационных технологий

Группа/Group K34422

Направление подготовки/ Subject area 45.03.04 Интеллектуальные системы в
гуманитарной сфере

Образовательная программа / Educational program Интеллектуальные системы в
гуманитарной сфере 2018

Язык реализации ОП / Language of the educational program Русский

Статус ОП / Status of educational program

Квалификация/ Degree level Бакалавр

Тема ВКР/ Thesis topic Разработка программного обеспечения с элементами анализа
данных для образовательного учреждения.

Руководитель ВКР/ Thesis supervisor Гусарова Наталия Федоровна, старший научный
сотрудник, кандидат технических наук, Университет ИТМО, факультет
инфокоммуникационных технологий, доцент (квалификационная категория "ординарный
доцент")

**ХАРАКТЕРИСТИКА ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЫ
DESCRIPTION OF THE GRADUATION THESIS**

Цель исследования / Research goal

Разработка и интеграция в интеллектуальную информационную систему учреждения
«Техническая школа» ГУП «Петербургский метрополитен» панели для анализа данных
студентов и модуля экспорта данных в полноценный документ.

Задачи, решаемые в ВКР / Research tasks

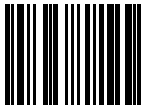
Выявление требований к панели для анализа данных студентов. Разработка и интеграция
панели для анализа данных студентов. Разработка и интеграция модуля экспорта данных.

Краткая характеристика полученных результатов / Short summary of results/findings

Были выявлены требования к панели для анализа данных студентов. По выявленным
требованиям была разработана и интегрирована в систему панель для анализа данных
студентов. Была разработана и интегрирована модуль экспорта данных в полноценный
документ. Было создано визуальное представление производственного календаря в
системе. Был автоматизирован импорт данных производственного календаря в систему.
Была реализована возможность авторизации в системе студентам.

Обучающийся/Student

Документ	
----------	--

подписан	
Кириллов Николай Александрович	
23.05.2022	

(эл. подпись/ signature)

Кириллов
Николай
Александрович

(Фамилия И.О./ name
and surname)

Руководитель ВКР/
Thesis supervisor

Документ подписан	
Гусарова Наталия Федоровна	
23.05.2022	

(эл. подпись/ signature)

Гусарова
Наталия
Федоровна

(Фамилия И.О./ name
and surname)

СОДЕРЖАНИЕ

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ.....	7
ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ.....	8
ВВЕДЕНИЕ.....	9
1 ТЕСТИРОВАНИЕ СИСТЕМЫ.....	11
1.1 Функциональное тестирование.....	11
1.2 Выводы по главе 1.....	14
2 АНАЛИТИЧЕСКАЯ ПАНЕЛЬ	15
2.1 Сбор данных	15
2.2 Визуализация собранных данных.....	16
2.3 Сравнительный анализ	17
2.4 Создание аналитической панели	20
2.5 Выводы по главе 2.....	35
3 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	36
3.1 Модуль экспорта отчетов	36
3.2 Создание возможности авторизации студента.....	39
3.3 Визуальное представление производственного календаря	40
3.4 Выводы по главе 3.....	45
ЗАКЛЮЧЕНИЕ	46
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	48

СПИСОК СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

БД – База данных

ГУП – Государственное унитарное предприятие

ИИС – Интеллектуальная Информационная Система

ПО – Программное обеспечение

ТШ – Техническая школа

CSV – Comma-Separated Values

JPEG – Joint Photographic Experts Group

JSON – JavaScript Object Notation

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

Дашборд – Аналитическая панель

Скрипт – это небольшая программа, созданная для автоматического выполнения задачи и состоящая из последовательности действий

Фреймворк – программное обеспечение, которое определяет структуру программной системы и облегчает разработку и объединение различных частей большого программного проекта

ВВЕДЕНИЕ

В современности хранение данных и их использование неразрывно связано с применением информационных технологий. Можно значительно повысить эффективность бизнес-процессов организации, внедрив в нее интеллектуальную информационную систему (ИИС) [1]. Необходимо упорядоченное, структурированное и целостное хранение информации для того, чтобы была возможность без труда обратиться к имеющейся информации, особенно при наличии нескольких различных информационных потоков.

Идеальным примером организации, нуждающейся в оптимизации информационных процессов, является «Техническая школа» ГУП «Петербургский метрополитен», которая осуществляет образовательную деятельность по реализации профессионального обучения по основным программам профессиональной подготовки, переподготовки, повышения квалификации и профессионального обучения по дополнительным программам переподготовки и повышения квалификации работников метрополитена.

Техническая школа прекрасно понимает описанную выше проблему и стремится ее решить. Именно поэтому, в настоящее время, ведется разработка интеллектуальной информационной системы для технической школы (ТШ).

Более того, ТШ остро нуждается в элементах учебной аналитики [2]. Такая потребность во многом обусловлена устройством образовательного процесса школы. Количество образовательных мест в технической школе всегда равно количеству требующихся в будущем машинистов электропоездов. В противном случае, если обучающиеся отчисляются, организация снова остается без необходимых квалифицированных кадров, а также вынуждена применять меры для возмещения с отчислившихся деньги за проведенное обучение. Следовательно, для обеспечения всех должностей необходимыми кадрами, требуется, чтобы каждый из студентов успешно закончил обучение и устроился на работу по полученной специальности.

Необходимо иметь возможность выявлять потенциально отчислившихся студентов, для дальнейшей работы с ними, чтобы избежать недобора сотрудников.

Для выявления отстающих в обучении студентов и дальнейшей им помощи необходимо анализировать учебный процесс. В качестве инструмента для анализа учебного процесса, который можно встроить в ИИС, идеально подойдет визуальное представление данных в виде интерактивной аналитической панели, или, как ее еще называют, дашборда. Интерактивную аналитическую панель можно удобно встроить в уже разрабатываемую интеллектуальную информационную систему.

Также, в процессе организации учебного процесса в технической школе есть необходимость создавать отчеты в виде документов под подпись, поэтому была поставлена задача разработать модуль в ИИС для автоматического экспорта данных в полноценный документ для печати и дальнейшего редактирования.

Таким образом задачами выпускной квалификационной работы являлись:

- 1) выявление требований, разработка и интеграция инструмента для анализа данных;
- 2) разработка и интеграция дополнительных модулей в систему:
 - модуль экспорта отчетов;
 - возможность авторизации студента;
 - визуальное представление производственного календаря;
- 3) проведение функционального тестирования разработанной системы.

1 ТЕСТИРОВАНИЕ СИСТЕМЫ

1.1 Функциональное тестирование

Полученная система нуждалась в проверке на функционирование, так как никакого тестирования не проводилось.

Было проведено функциональное тестирование.

Функциональное тестирование – это тестирование программного обеспечения с целью проверки реализации функциональных требований, другими словами, способности ПО в определённых условиях решать пользовательские задачи.

В качестве тестирования был определен сценарий использования, затрагивающий наибольшее количество модулей системы, таким сценарием является процесс заполнения электронного журнала преподавателем.

Шаги, которые необходимо сделать, для заполнения электронного журнала в системе:

- 1) получение календарного плана;
- 2) добавление календарного плана;
- 3) добавление в систему производственного календаря;
- 4) создание в системе подразделения;
- 5) создание в системе преподавателя:
 - создание в системе пользователя;
 - создание в системе персональных данных сотрудника;
 - создание в системе сотрудника;
 - создание в системе преподавателя;
- 6) создание в системе студента:
 - создание в системе персональных данных студента;
 - создание в системе студента;
- 7) создание в системе учебной группы;
- 8) добавление студента в группу;
- 9) добавление учебной программы;
- 10) создание в системе предмета;

- 11) добавление предмета в программу;
- 12) создание в системе учета часов;
- 13) создание в системе занятия;
- 14) отметка посещения;
- 15) добавление оценки.

Все вышеперечисленные шаги были пройдены, была обнаружена только одна ошибка. Обнаруженная ошибка заключается в невозможности создания занятия (рисунок 1).

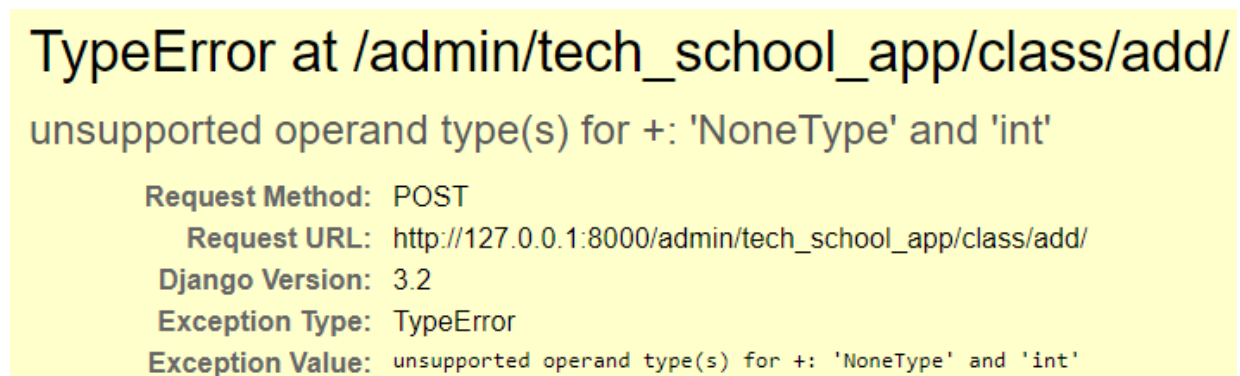


Рисунок 1 – Ошибка в создании занятия

Ошибка заключалась в невозможности провести математическую операцию между объектами `int` и `None` и находилась в функции проверки на лимит часов. Ошибка возникает в момент, когда переменная `classes` является пустой, то есть, когда других занятий, помимо создаваемого в системе еще нет.

Было добавлено условное выражение, проверяющее `classes` на наличии объектов внутри, и присваивающее `0` в случае их отсутствия. В результате, вместо объекта `None` будет объект `int`, равный нулю, что позволит без ошибок выполнить программный код.

Исправленный программный код можно увидеть на рисунке 2, жёлтым выделен фрагмент с добавленным кодом.

```
def hours_match_limit(self):  
    """  
    Проверяет, что сумма часов добавляемых занятий не превышает лимит из "учета часов" Возвращает Boolean.  
    """  
    _date = self.when.date()  
    _group = self.group  
    _teacher = self.teacher  
    _subject = self.subject  
    th = TrainingHours.objects.filter(group=_group, teacher=_teacher, subject=_subject, date=_date)  
    classes = Class.objects.filter(when=self.when)  
    hours_th = th.aggregate(Sum('hours'))["hours__sum"]  
    hours_classes = classes.aggregate(Sum('hours'))["hours__sum"] if len(classes) != 0 else 0
```

Рисунок 2 – Исправленный программный код

Также был выполнен перевод пользовательского интерфейса на русский язык в тех местах, где он отсутствовал. Пример переведенных элементов представлен на рисунках 3–6.

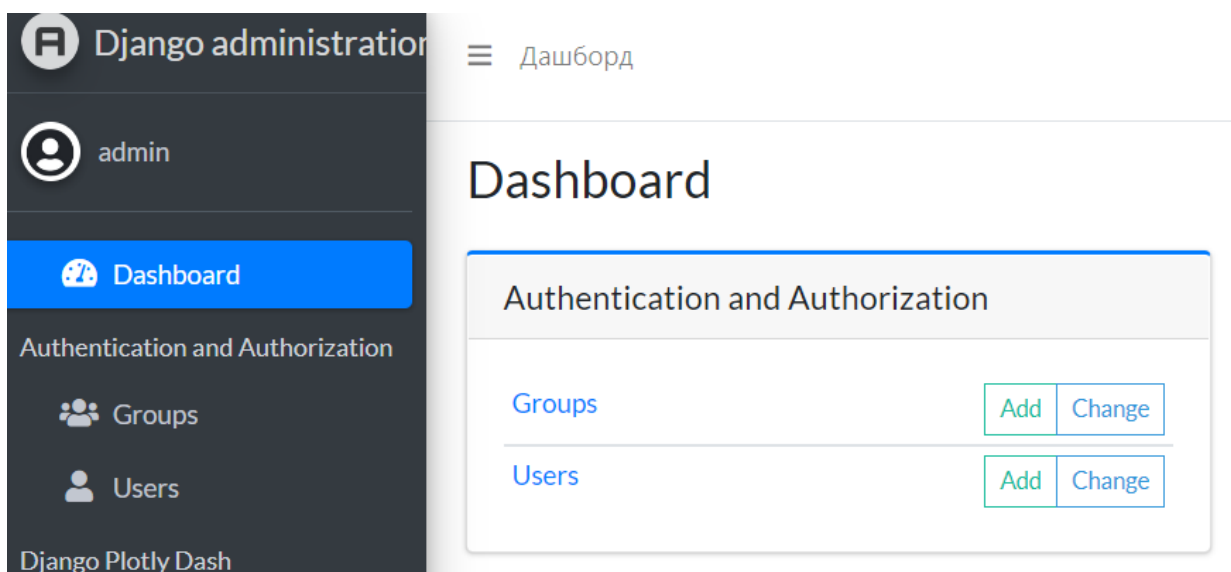


Рисунок 3 – Фрагмент интерфейса 1 до перевода

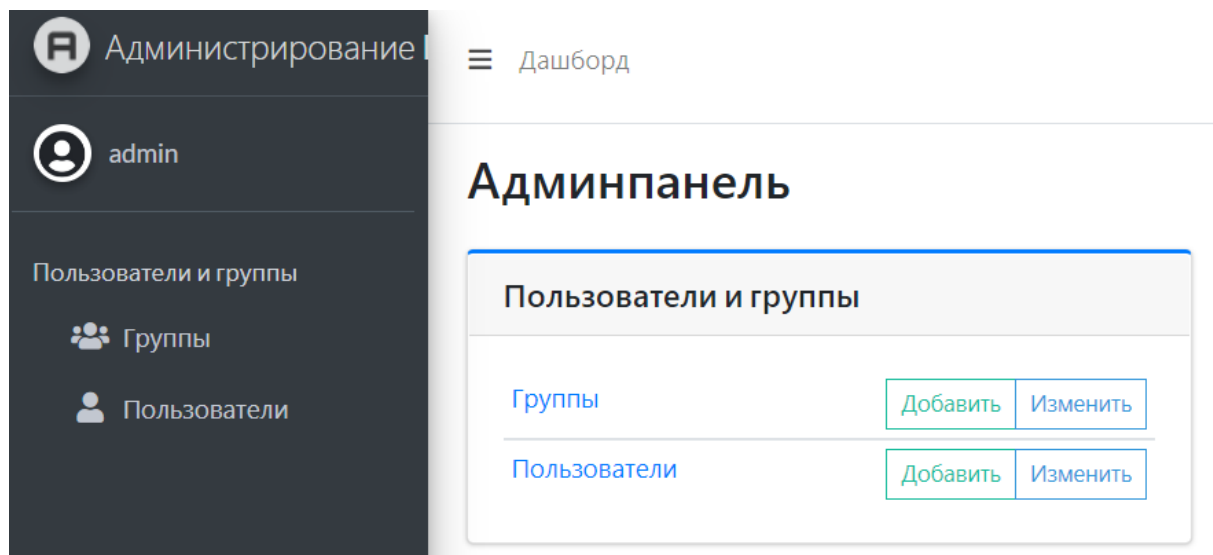


Рисунок 4 – Фрагмент интерфейса 1 после перевода

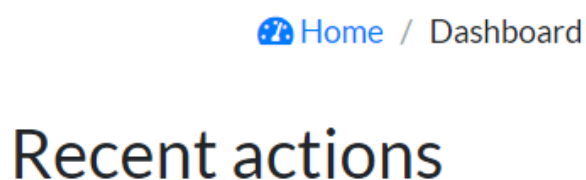


Рисунок 5 – Фрагмент интерфейса 2 до перевода

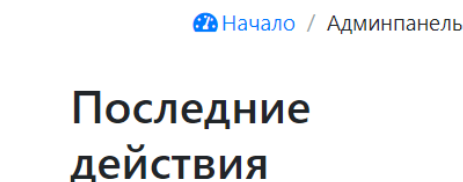


Рисунок 6 – Фрагмент интерфейса 2 после перевода

1.2 Выводы по главе 1

Был определен пользовательский сценарий, затрагивающий наибольшее количество модулей системы, заключающийся в ведении электронного журнала.

По определенному сценарию было проведено функциональное тестирование системы.

В результате проведенного функционального тестирования системы была выявлена ошибка, возникающая во время создания.

Выявленная ошибка была исправлена.

Была проведена локализация некоторых элементов системы.

2 АНАЛИТИЧЕСКАЯ ПАНЕЛЬ

2.1 Сбор данных

Для возможности просмотра и сравнения данных, отслеживания их тенденций, за прошедшие года, хранящихся в бумажном формате, необходимо привести их к цифровому машиночитаемому формату.

Сотрудниками Технической школы были предоставлены данные ведомостей обучающихся с текущими оценками по всем предметам и оценками по экзаменам. Данные были предоставлены в виде фотографий ведомостей с 2009 по 2019 года.

Для возможности просмотра и сравнения данных, отслеживания их тенденций, за прошедшие года, хранящихся в бумажном формате, необходимо привести их к цифровому машиночитаемому формату. Данные были перенесены в формат CSV. Главная причина выбора CSV формата заключается в его широкой совместимости. Файл CSV можно открыть практически в любом приложении, или даже в текстовом редакторе, так как сам формат является текстовым файлом, в котором используются специальные символы, к примеру запятая или точка с запятой, для определения того, где заканчивается одна ячейка, а другая начинается. CSV файлы широко используются в аналитике, так что использование именно такого формата обеспечивает возможность использования этих файлов для анализа данных в будущем.

В качестве примера предоставленной ведомости приведен фрагмент с намеренно замазанными элементами для обеспечения конфиденциальности информации.

Фрагмент ведомости представлен на рисунке 7.

№ п/п	Фамилия и имена учащихся	Месяцы		Оценки									
		Предметы	Итого	ПТ	О.Т.	М.Т.	М.Т.	М.Т.	М.Т.	М.Т.	М.Т.	М.Т.	М.Т.
1	[REDACTED]												
2	[REDACTED]												
3	[REDACTED]												
4	[REDACTED]												
5	[REDACTED]												

№ п/п	Фамилия и имена учащихся	Месяцы		Оценки									
		Предметы	Итого	ПТ	О.Т.	М.Т.	М.Т.	М.Т.	М.Т.	М.Т.	М.Т.	М.Т.	М.Т.
1	[REDACTED]												
2	[REDACTED]												
3	[REDACTED]												
4	[REDACTED]												
5	[REDACTED]												

Рисунок 7 – Фрагмент ведомости с оценками

2.2 Визуализация собранных данных

По полученным данным были визуализированы оценки по нескольким предметам. Примеры построенных визуализаций приведены на рисунках 8 и 9.

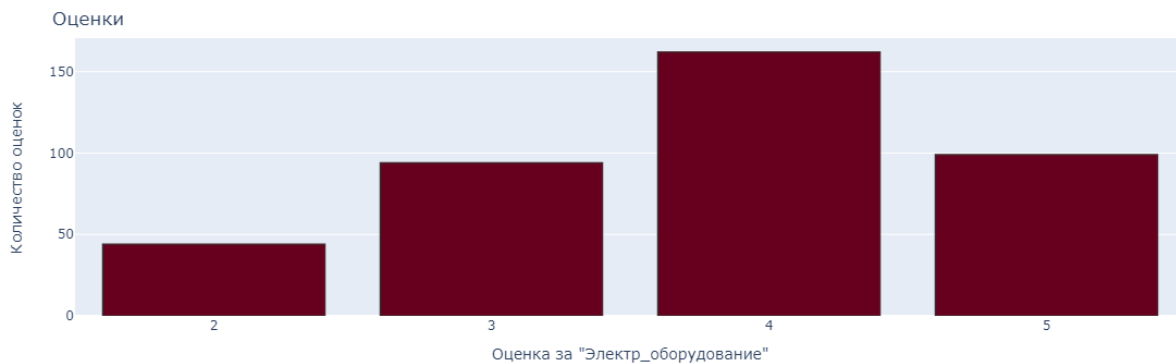


Рисунок 8 – Столбиковая диаграмма распределения оценок по предмету «Электронное оборудование»

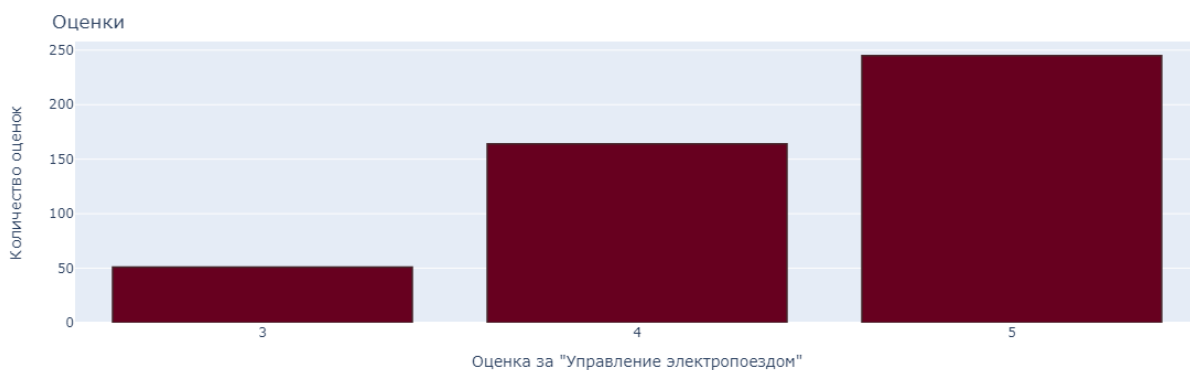


Рисунок 9 – Столбиковая диаграмма распределения оценок по предмету «Управление электропоездом»

Также была построена кольцевая диаграмма, показывающая отношение успешно окончивших обучение студентов к отчисленным.

Построенная кольцевая диаграмма приведена на рисунке 10.

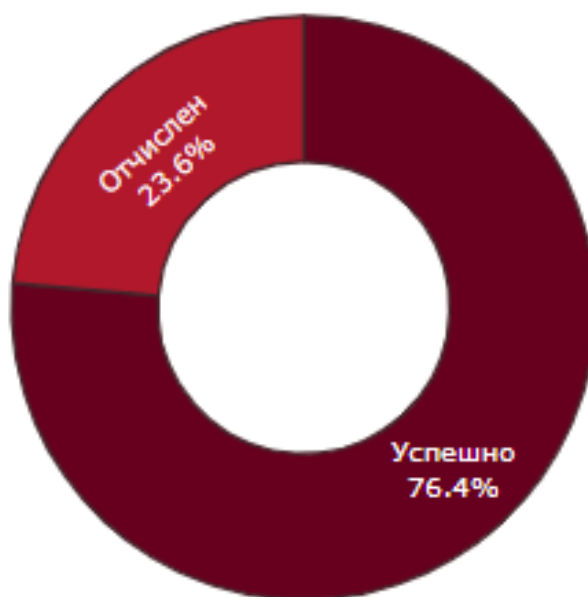


Рисунок 10 – Кольцевая диаграмма распределения отчисления студентов

Можно сделать вывод, что хоть и большинство студентов (76.4%) успешно оканчивают свое обучение, но также присутствует немалый процент выбывших студентов (23.6%).

Это еще раз доказывает необходимость создания инструмента для аналитики данных в технической школе.

2.3 Сравнительный анализ

В качестве систем для создания аналитической панели для сравнительного анализа были выбраны следующие продукты:

- Yandex DataLens [3],
- Tableau [4],
- Power BI [5],
- Qlik Sense [6].

Yandex DataLens – это российский сервис для бизнес-аналитики, сочетающий в себе возможности подключения к различным источникам данных, визуального их представления и сбора построенных визуализаций в аналитические панели.

Tableau – это американское программное обеспечение, применяемое для интерактивной бизнес-аналитики и визуализации данных. Tableau обладает возможностью взаимодействия с различными облачными сервисами, такими как Dropbox, Google Таблицы, AWS Redshift и другими.

Power BI – это американское комплексное программное обеспечение от компании Microsoft для бизнес-анализа, включающее в себе несколько программных продуктов, а также web-сервисов, ключевой продукт – Power BI Desktop.

Qlik Sense – американский продукт, предназначенный для визуализации данных и проведения исследований, позволяющий создавать гибкие интерактивные визуализации, приводящие к обоснованным выводам.

Итоги сравнительного анализа приведены в таблице 1.

Таблица 1 – Сравнение аналогов

Критерий	Yandex DataLens	Tableau	Power BI	Qlik Sense
Наличие бесплатной версии	Да	Нет	Да	Нет
Кроссплатформенность	Браузерное приложение	Да	Да	Браузерное приложение
Наличие инструкций/документаций (наличие обучающих материалов)	Да	Да	Да	Да
Наличие уровней доступа	Да	Да	Да	Да
Поддерживаемая операционная система	Браузерное приложение	MS Win 8/8.1, Win 10 (x64), macOS Mojave 10.14, macOS Catalina 10.15, и Big Sur 11.4 или новее	MS Windows	Браузерное приложение
Автосохранение	Да	Да	Да	Да
Контроль версий	Нет	Да	Нет	Нет
Отечественный сервис	Да	Нет	Нет	Нет
Возможность интеграции в Django проект	Не найдено	Да	Да	Не найдено

Сравнительный анализ аналогичных решений показал, что все они, хоть и обладают разнообразным функционалом, но подходящего решения, которое сочетало бы в себе все необходимые функции, нет. Большая часть из рассмотренных продуктов не являются отечественными, что неприемлемо для технической школы, так как необходимо, чтобы программное решение не зависело от иностранных компаний. Более того, половину из них можно использовать исключительно как отдельный продукт, без возможности встраивания их в уже разрабатываемую систему.

На основании исследования существующих программных решений было принято решение разработать свое, независимое программное решение.

2.4 Создание аналитической панели

Смысл аналитических панелей заключается в том, что на одном экране располагаются все необходимые метрики. С помощью этих метрик можно выявить и проанализировать тенденции и изменения.

Визуализация данных – это один из вариантов их представления, который обеспечивает работу по их дальнейшему изучению наиболее эффективным образом. В свою очередь, визуализация данных широко применяется в различных научных и статистических исследованиях.

Создание и интеграция интерактивной аналитической панели в разрабатываемую систему для ТШ обеспечит удобное проведение анализа обучения студентов.

Были выдвинуты следующие требования к аналитической панели:

- 1) должна быть возможность фильтрации данных по времени и группам,
- 2) должна присутствовать визуализация оценок студентов,
- 3) должна присутствовать визуализация посещаемости занятий.

Было принято решение, в качестве средства по визуализации данных использовать `django-plotly-dash` [7]. Использование `django-plotly-dash` позволяет использовать приложения Plotly Dash [8] в приложениях Django. Выбор использовать `django-plotly-dash` был сделан, так как интеграция модуля

с аналитической панелью происходит в систему, написанную с использованием фреймворка Django [9].

Dash – это фреймворк для быстрого создания приложений для работы с данными, написанный на основе Plotly.js и React.js, Dash идеально подходит для создания и развертывания приложений для работы с данными с настраиваемыми пользовательскими интерфейсами.

Фреймворк – программное обеспечение, которое определяет структуру программной системы и облегчает разработку и объединение различных частей большого программного проекта.

Для построения графиков необходимые таблицы из базы данных были преобразованы в dataframe с помощью pandas. Был создан общий набор данных df и созданы два графика.

Код создания набора данных и графиков приведены на рисунке 11.

```
df_grades = pd.DataFrame(grades)
df_groups = pd.DataFrame(groups)
df_membership = pd.DataFrame(membership)
df_incp = pd.DataFrame(incp)
df_calplan = pd.DataFrame(calplan)
df_class = pd.DataFrame(_class)

df_personalinfo = pd.DataFrame(personalinfo)
df_personalinfo['FullName'] = df_personalinfo['surname'] + ' ' + df_personalinfo['name'] + ' ' + df_personalinfo[
    'patronymic']
df_students = pd.DataFrame(students)

df = pd.merge(df_grades[['grade_type', 'grade', 'attendance', 'student_id', 'class_id_id']],
              df_membership[['group_id', 'student_id']],
              left_on='student_id', right_on='student_id')
df = pd.merge(df, df_class[['id', 'when']], left_on='class_id_id', right_on='id')
df = pd.merge(df, df_groups[['id', 'name', 'program_id']], left_on='group_id', right_on='id')
df = pd.merge(df, df_students[['id', 'personal_info_id', 'personnel_num']], left_on='student_id', right_on='id')
df = pd.merge(df, df_personalinfo[['id', 'FullName']], left_on='personal_info_id', right_on='id')
df['when'] = pd.to_datetime(df['when']).dt.date
students_list = df[['personnel_num', 'group_id', 'FullName']].drop_duplicates(subset=['personnel_num'])
students_list = students_list.rename(columns={"personnel_num": "Табельный номер", "FullName": "ФИО"})
df_g = df[df["grade_type"] == 'g']
fig_grades = px.bar(df_g, x=df_g['grade'].value_counts().index, y=df_g['grade'].value_counts().values.tolist(),
                    title='Оценки')
pie_attendance = px.pie(df[df["grade_type"] == 'a'], names="attendance", title="Процент посещений", hole=0.5)
```

Рисунок 11 – Код создания графиков и необходимых наборов данных

Для построения графиков использовался модуль `plotly.express` [10] (импортирован как `px`). Он содержит функции, которые могут создавать целые фигуры сразу. Plotly Express является встроенной частью библиотеки Plotly.

Dash приложения состоят из двух частей.

Первая часть – это макет приложения, описывающий то, как приложение будет выглядеть.

Вторая часть описывает интерактивность приложения с помощью функций обратного вызова.

После того, как были подготовлены графики, следует создание макета аналитической панели.

Был реализован фильтр, содержащий поле с выбором группы, с помощью метода Dash Core Components Dropdown [11].

Код реализации фильтрации приведен на рисунке 12.

```
app.layout = html.Div(
    children=[
        html.Div(children='Выберите группу',
                  style={'display': 'block', 'fontSize': "24px", 'margin-bottom': '5px', 'margin-top': '0px',
                        'color': 'rgb(42, 63, 95)'},
                  className='menu-title'),
        html.Div(
            dcc.Dropdown(
                id='group-filter',
                options=[
                    {'label': group_name, 'value': group}
                    for group in df.group_id.unique()
                    for group_name in df[df['group_id'] == group].name.unique()
                ],
                className='dropdown',
                placeholder='Группа',
            ), style={'width': '100%', 'display': 'block', 'margin-bottom': '5px', 'vertical-align': 'top'})
    ]
)
```

Рисунок 12 – Код реализации фильтрации

Внешний вид созданного фильтра представлен на рисунке 13.

Выберите группу

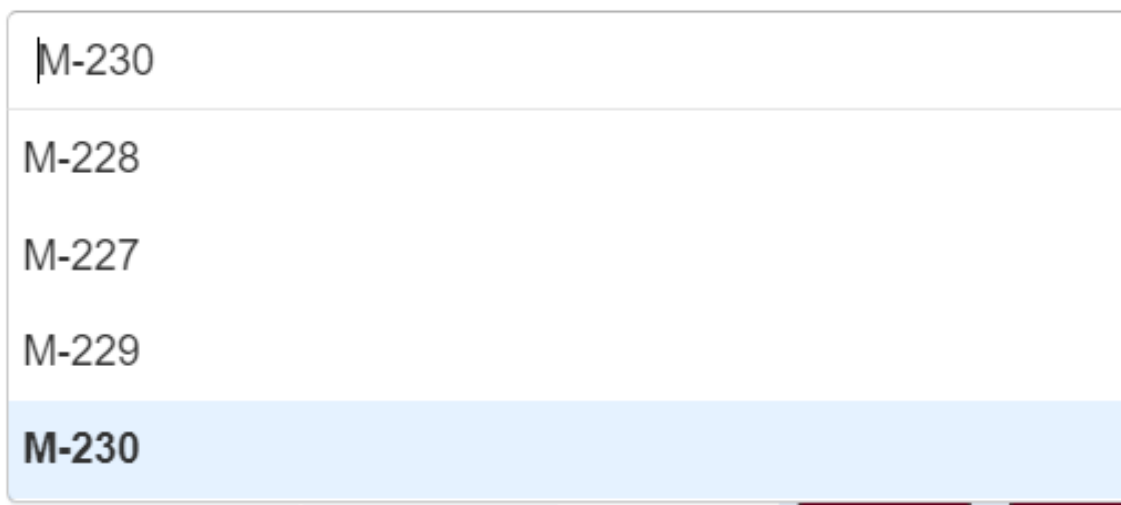


Рисунок 13 – Созданный фильтр

Далее была реализована возможность выбора временного промежутка для отображаемых данных с помощью метода Dash Core Components `DatePickerRange` [12].

Код реализации фильтра по датам представлен на рисунке 14.

```
html.Div(  
    dcc.DatePickerRange(  
        id='my-date-picker-range',  
        display_format='DD.MM.YYYY',  
        start_date_placeholder_text='Начальная дата',  
        end_date_placeholder_text='Конечная дата',  
        start_date=date.today(),  
        end_date=date.today(),  
        first_day_of_week=1,  
        show_outside_days=True,  
        number_of_months_shown=2,  
    ), style={'width': '100%', 'display': 'block', 'margin-bottom': '5px'}
```

Рисунок 14 – Код реализации фильтрации по датам

Внешний вид созданного фильтра по датам представлен на рисунке 15.

Как можно увидеть, выбор дат происходит путем выбора промежутка дат в визуальном представлении календаря, что является удобным решением.

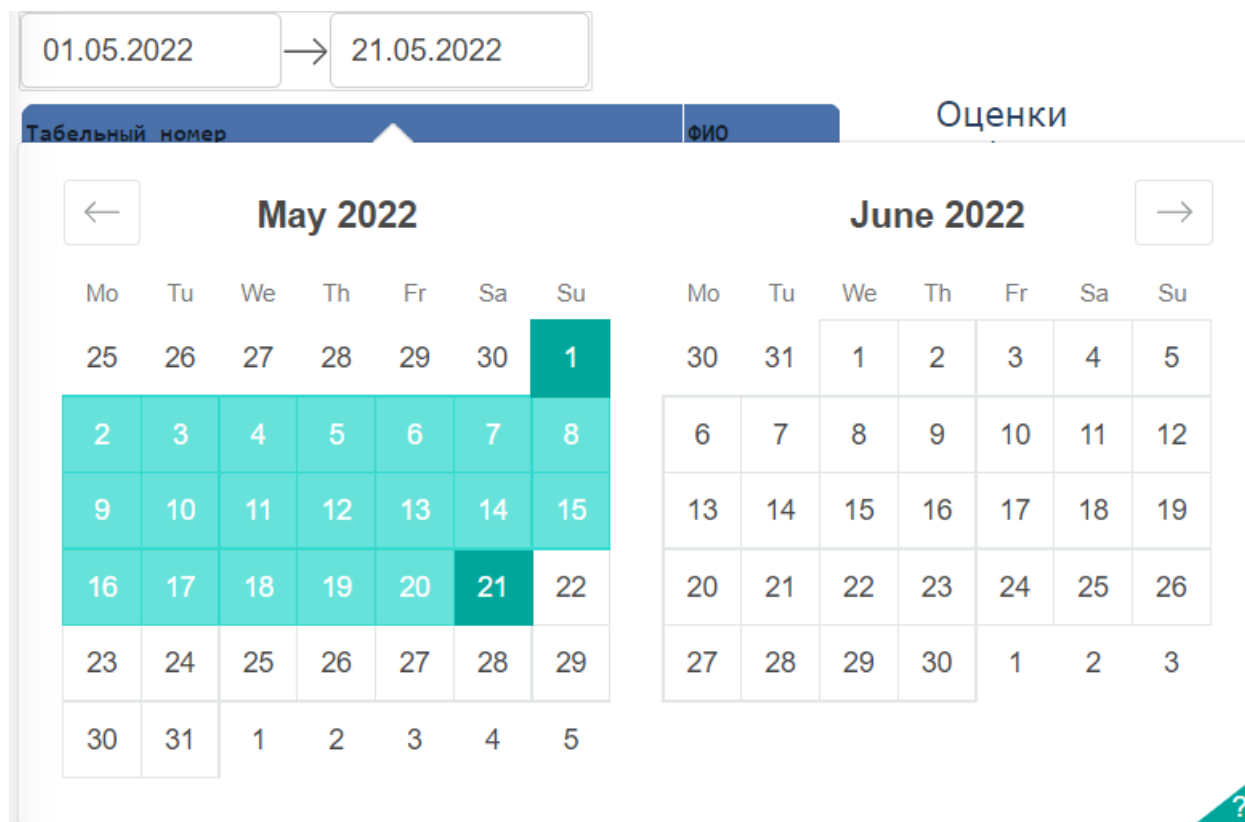


Рисунок 15 – Внешний вид фильтрации по датам

Так как разрабатываемая аналитическая панель разрабатывается для возможности анализировать обучающихся, было принято решение, в качестве одной из визуализаций вывести таблицу со студентами выбранной группы. В качестве метода реализации таблицы использовался Dash DataTable [13].

Код реализации визуального представления списка студентов в виде таблицы именами студентов и их группами приведен на рисунке 16.

```
html.Div(
    dash_table.DataTable(
        id='table',
        columns=[{"name": i, "id": i} for i in students_list[['Табельный номер', 'ФИО']]],
        data=df.to_dict('records'),
        style_cell_conditional=[
            {
                'textAlign': 'left',
            }
        ],
        style_table={
            'border': '1px solid white',
            'borderRadius': '10px',
            'overflow': 'hidden'
        },
        style_header={
            'border': '1px solid white',
            'backgroundColor': 'rgb(75, 113, 171)',
            'color': 'rgb(19, 29, 44)',
            'fontWeight': 'bold'
        },
        style_data={
            'border': '1px solid white',
            'color': 'rgb(42, 63, 95)',
            'backgroundColor': 'rgb(229, 236, 246)'
        },
    ), style={'width': '30%', 'display': 'inline-block', 'vertical-align': 'top'}
```

Рисунок 16 – Код реализации таблицы со студентами

В качестве визуализируемых параметров были выбраны табельные номера студентов и их полные имена.

Внешний вид полученной таблицы приведен на рисунке 17.

Табельный номер	ФИО
00034	Грибачев Сергей Викторович
00035	Романов Антон Сергеевич
00036	Лебедев Александр Александрович
00037	Купцов Сергей Олегович
00038	Харитонов Михаил Александрович
00039	Баранов Дмитрий Александрович
00040	Носков Семен Сергеевич

Рисунок 17 – Внешний вид таблицы со студентами

В качестве диаграммы для визуализации оценок студентов был выбрана столбиковая диаграмма. Классическая столбиковая диаграмма оперирует горизонтальными или вертикальными столбцами, демонстрируя дискретные, числовые сравнения между разными категориями. Одна ось диаграммы представляет конкретные сравниваемые категории, а другая – шкала дискретных значений.

В качестве диаграммы для визуализации посещаемости студентов была выбрана кольцевая диаграмма. Кольцевая диаграмма – это круговая диаграмма, только с вырезанной центральной частью. Круговые диаграммы часто встречаются в презентациях, либо в офисной документации. Они хорошо отображают пропорциональное и процентное соотношение между какими-либо категориями, разделяя круг на пропорциональные сегменты. Длины полученных в результате деления круга дуг представляют собой пропорциональные доли каждой из категорий, в то время как целый круг представляет общую сумму всех данных, эквивалентную 100%.

У кольцевой диаграммы есть ощутимое преимущество перед обычной круговой диаграммой. Круговую диаграмму иногда критикуют за акцентирование внимания на относительных размерах частей по отношению друг ко другу и диаграмме в целом. Кольцевая диаграмма снижает фокус на отдельных областях, что позволяет наблюдателю сосредоточиться больше на изменениях общих значений, его внимание будет приковано к длине дуг, а не к сопоставлению пропорций между кусочками круга и тем самым решает проблему.

Код реализации графиков в макете панели приведен на рисунке 18.

```
html.Div(  
    children=[  
        html.Div(  
            children=dcc.Graph(  
                id='bar_1',  
                figure=fig_grades  
            ), style={'display': 'block'}  
        ),  
        html.Div(  
            children=dcc.Graph(  
                id='pie_attendance',  
                figure=pie_attendance  
            ), style={'display': 'block'}  
        )  
    ], style={'width': '70%', 'display': 'inline-block'}  
)
```

Рисунок 18 – Код реализации графиков на панели

Для того, чтобы аналитическая панель была интерактивной, а не представляла статичные графики, необходимо создать функции обратного вызова, функций, которые автоматически вызываются Dash при изменении свойства входного компонента, чтобы обновить какое-либо свойство в другом компоненте.

Код реализации функции обратного вызова для кольцевой диаграммы приведен на рисунке 19.

```
@app.callback(
    Output("pie_attendance", "figure"),
    Input('my-date-picker-range', 'start_date'),
    Input('my-date-picker-range', 'end_date'),
    Input("group-filter", "value"))
def update_charts(start_date, end_date, value):
    start_date_object = pd.to_datetime(start_date)
    end_date_object = pd.to_datetime(end_date)
    filtered_data = df[df["group_id"] == value]
    filtered_data = filtered_data[filtered_data["grade_type"] == 'a']
    filtered_data = filtered_data[
        (filtered_data["when"] > start_date_object) & (filtered_data["when"] < end_date_object)]
    filtered_data["attendance_viz"] = np.where(filtered_data["attendance"] == 0, 'Пропуск', 'Посещение')

    bar = px.pie(
        filtered_data,
        names='attendance_viz',
        title="Процент посещений",
        hole=0.5,
        labels={'attendance_viz': 'Статус'},
        color_discrete_sequence=px.colors.sequential.RdBu
    )
    bar.update_layout(margin=dict(t=25), showlegend=False)
    bar.update_traces(textinfo='percent+label', marker_line_width=1.5, marker_line_color='rgb(69, 38, 43)')
    return bar
```

Рисунок 19 – Код реализации функции обратного вызова для кольцевой диаграммы

На рисунках 20 и 21 представлены фрагменты кода реализации функции обратного вызова для столбиковой диаграммы с переопределением полей и форматированием данных для вывода имен студентов во всплывающем меню при наведении на элемент графика.

```

@app.callback(
    Output("bar_1", "figure"),
    Input('my-date-picker-range', 'start_date'),
    Input('my-date-picker-range', 'end_date'),
    Input("group-filter", "value"))
def update_charts(start_date, end_date, value):
    start_date_object = pd.to_datetime(start_date)
    end_date_object = pd.to_datetime(end_date)

    filtered_data = df[df["group_id"] == value]
    filtered_data = filtered_data[filtered_data["grade_type"] == 'g']
    filtered_data = filtered_data[
        (filtered_data["when"] > start_date_object) & (filtered_data["when"] < end_date_object)]

    grade_groups = filtered_data[["FullName", "grade"]].groupby('grade')['FullName'].apply(list)
    df_grade_groups = pd.DataFrame(grade_groups).reset_index()
    if not df_grade_groups['FullName'].empty:
        df_grade_groups['amount'] = df_grade_groups['FullName'].str.len()
        df_grade_groups['FullName'] = df_grade_groups['FullName'].astype(str)
        df_grade_groups['FullName'] = df_grade_groups['FullName'].str.replace('[]', '')
        df_grade_groups['FullName'] = df_grade_groups['FullName'].str.replace("'", "")
        df_grade_groups['FullName'] = df_grade_groups['FullName'].str.replace(',', ' ', <br>)
    else:
        df_grade_groups['amount'] = 0

```

Рисунок 20 – Код реализации функции обратного вызова для столбиковой диаграммы часть 1

```

bar = px.bar(
    df_grade_groups,
    x='grade',
    y='amount',
    hover_data=['FullName'],
    orientation='v',
    labels={'FullName': 'Студент(ы)',
            'grade': 'Оценка',
            'amount': 'Количество оценок',
            'student_id': 'ID',
            'personnel_num': 'Табельный номер'},
    title='Оценки',
    color_discrete_sequence=px.colors.sequential.RdBu
)
bar.update_layout(margin=dict(t=25))
bar.update_layout(legend=dict(font=dict(size=10)))
bar.update_traces(marker_line_color='rgb(69, 38, 43)',
                  marker_line_width=1.5)
bar.update_xaxes(dtick=1)
bar.update_yaxes(dtick=1)
return bar

```

Рисунок 21 – Код реализации функции обратного вызова для столбиковой диаграммы часть 2

Полученная столбиковая диаграмма распределения оценок по студентам приведена на рисунке 22.

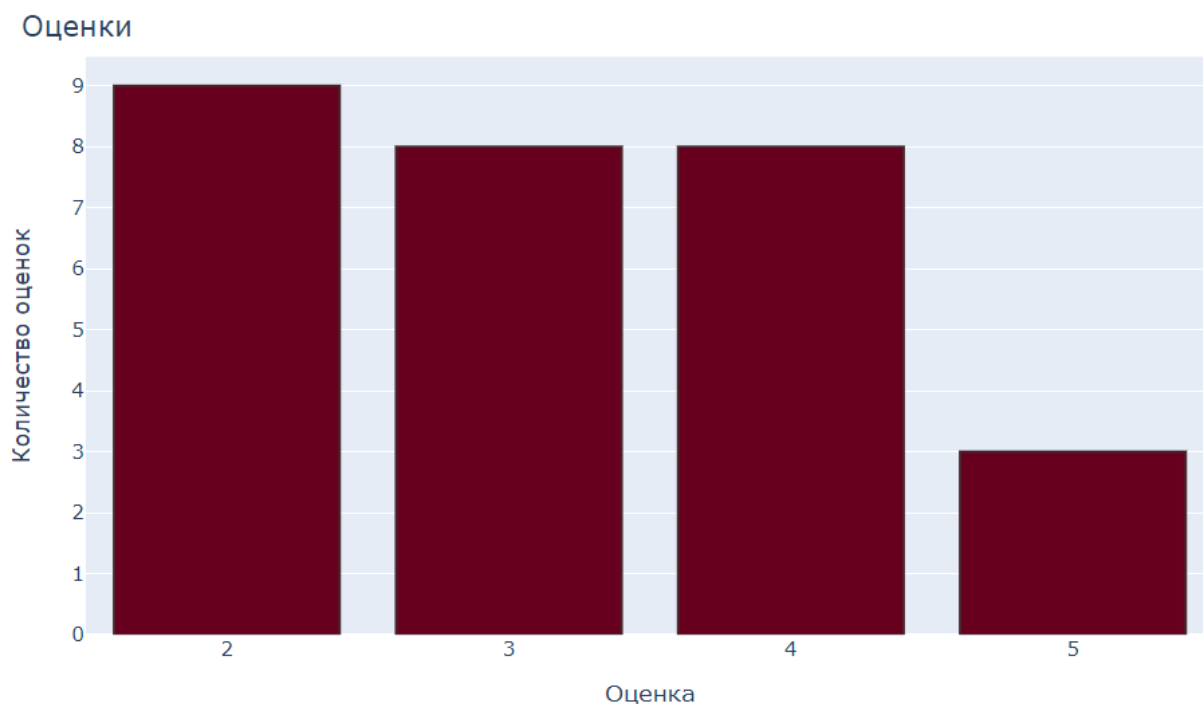


Рисунок 22 – Столбиковая диаграмма распределения оценок по студентам

При наведении курсора мыши на элемент диаграммы появляется всплывающее меню с дополнительной информацией, такой как: оценка, количество таких оценок и полные имена студентов, получивших эту оценку.

Пример описанного выше всплывающего меню приведен на рисунке 23.

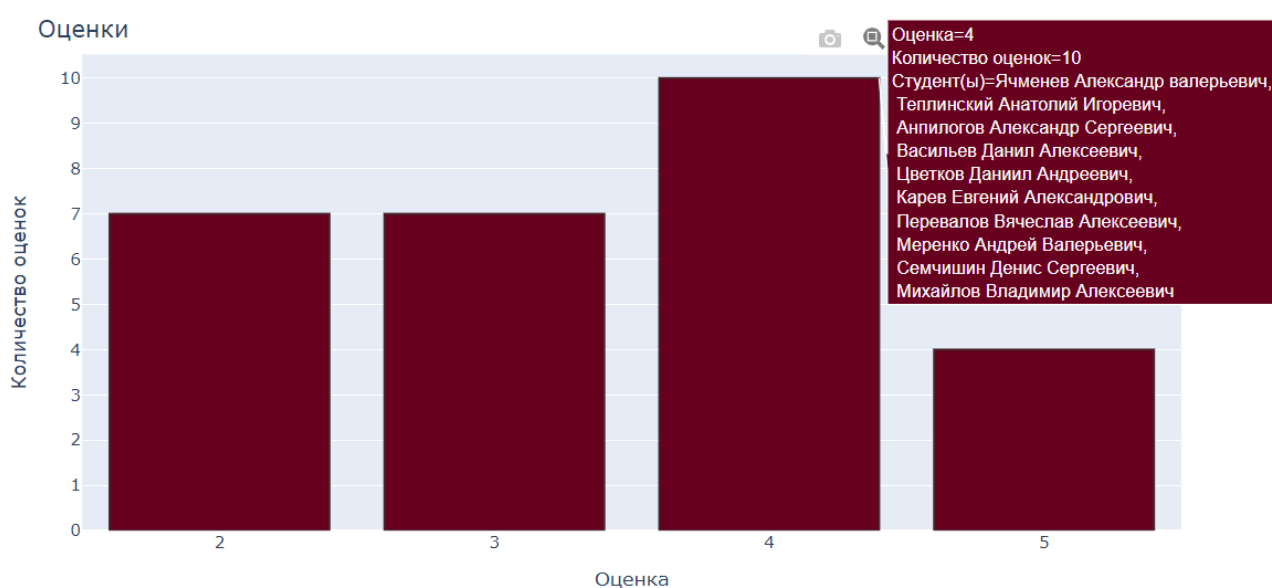


Рисунок 23 – Всплывающее меню с дополнительной информацией

Полученная кольцевая диаграмма распределения посещаемости студентов приведена на рисунке 24.

Процент посещений

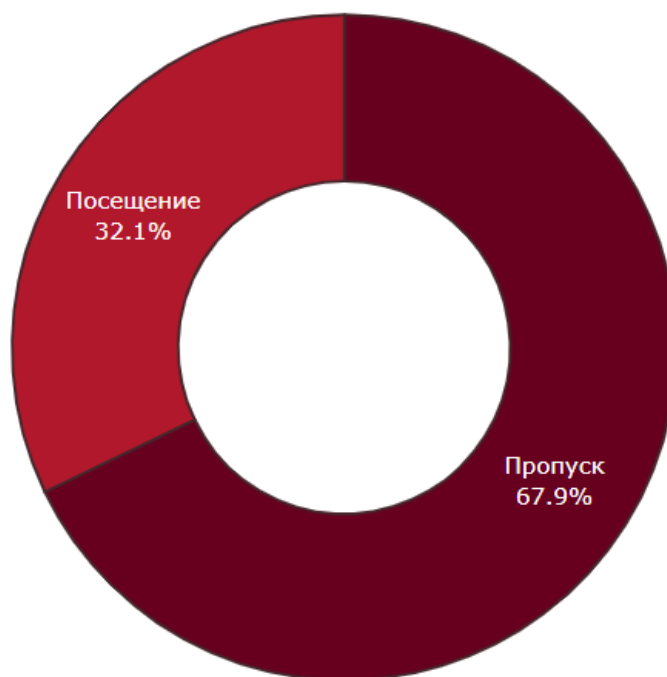


Рисунок 24 – Кольцевая диаграмма распределения посещаемости студентов

На рисунках 25 и 26, в качестве примеров, представлены аналитические панели для разных групп.

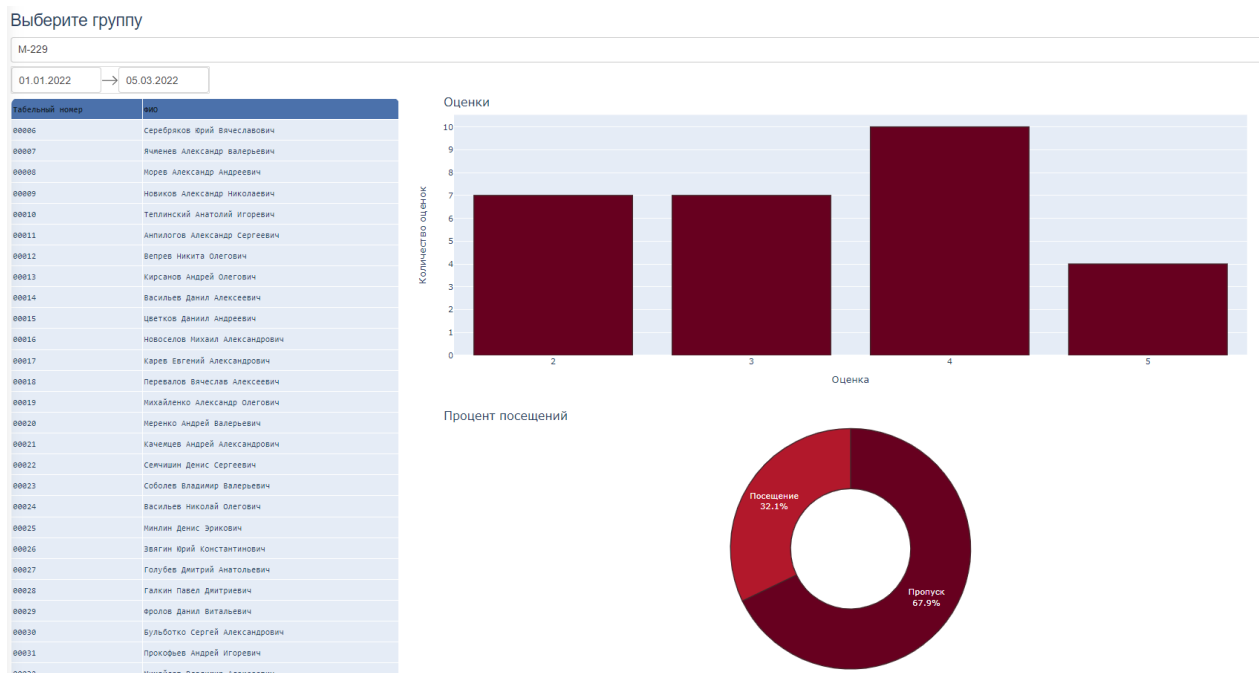


Рисунок 25 – Аналитическая панель с выбранной группой M-229

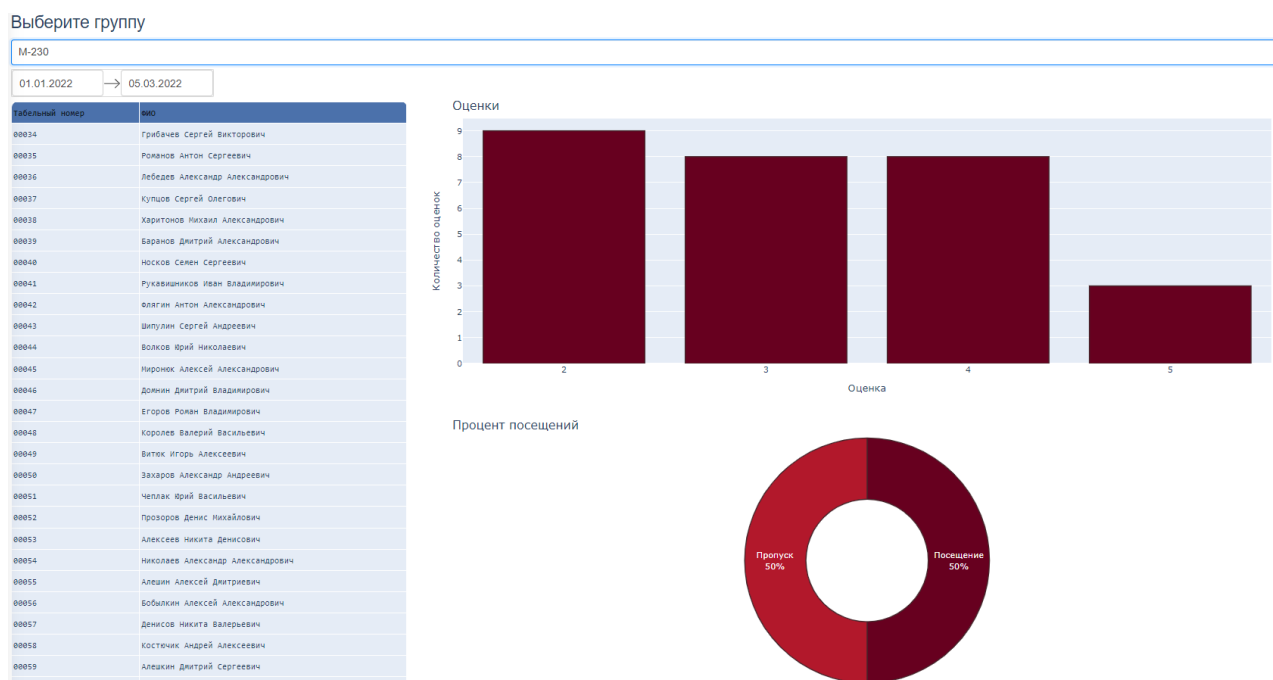


Рисунок 26 – Аналитическая панель с выбранной группой М-230

В случае, если пользователь не выбрал временной промежуток, либо в базе данных, во время выбранного временного промежутка отсутствуют данные, по какой-либо причине, то пользователь все равно увидит список группы, но поля графиков будут пустыми (рисунок 27).

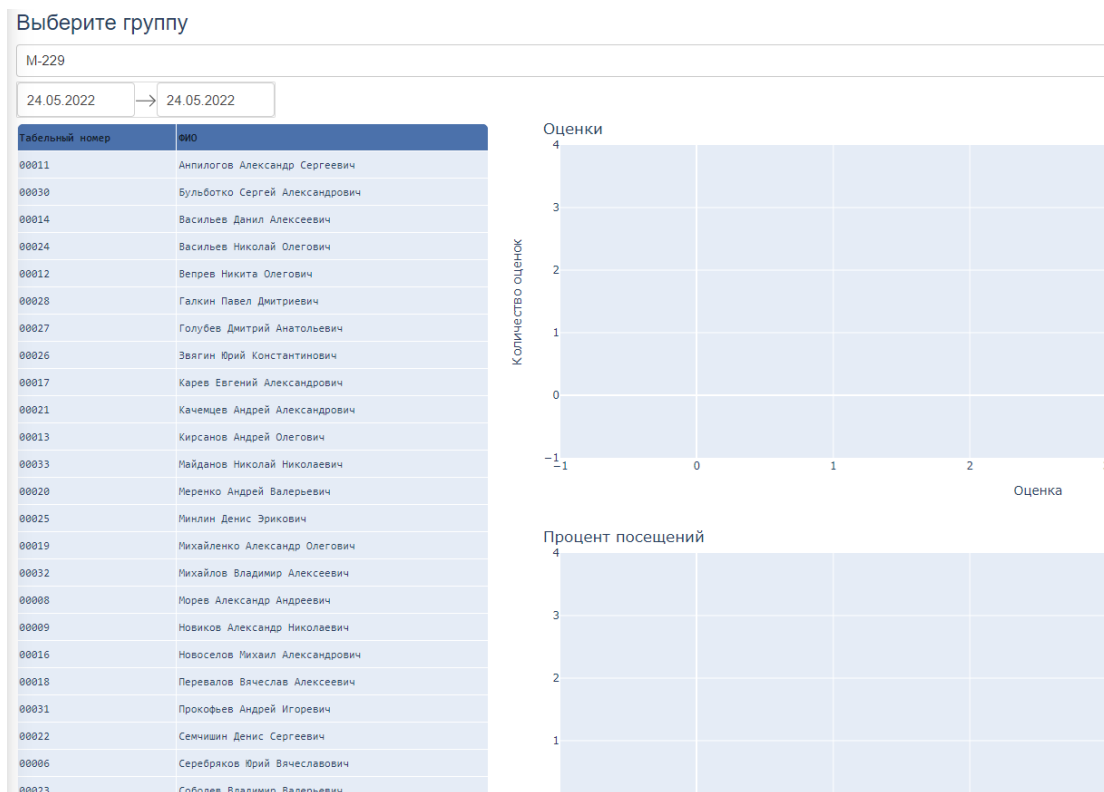


Рисунок 27 – Внешний вид аналитической панели, в случае отсутствия данных, либо невыбранного фильтра по времени

Так как наименования полей в базе данных записано на латинице, вся информация была русифицирована, для удобства использования.

Пример русификации с переопределением имен полей приведен на рисунке 28.

```
labels={'FullName': 'ФИО студента',  
        'grade': 'Оценка',  
        'student_id': 'ID студента',  
        'personnel_num': 'Табельный номер'},
```

Рисунок 28 – Русификация визуальной части

При желании графики можно масштабировать. Пример с масштабированием графика приведен на рисунке 29.

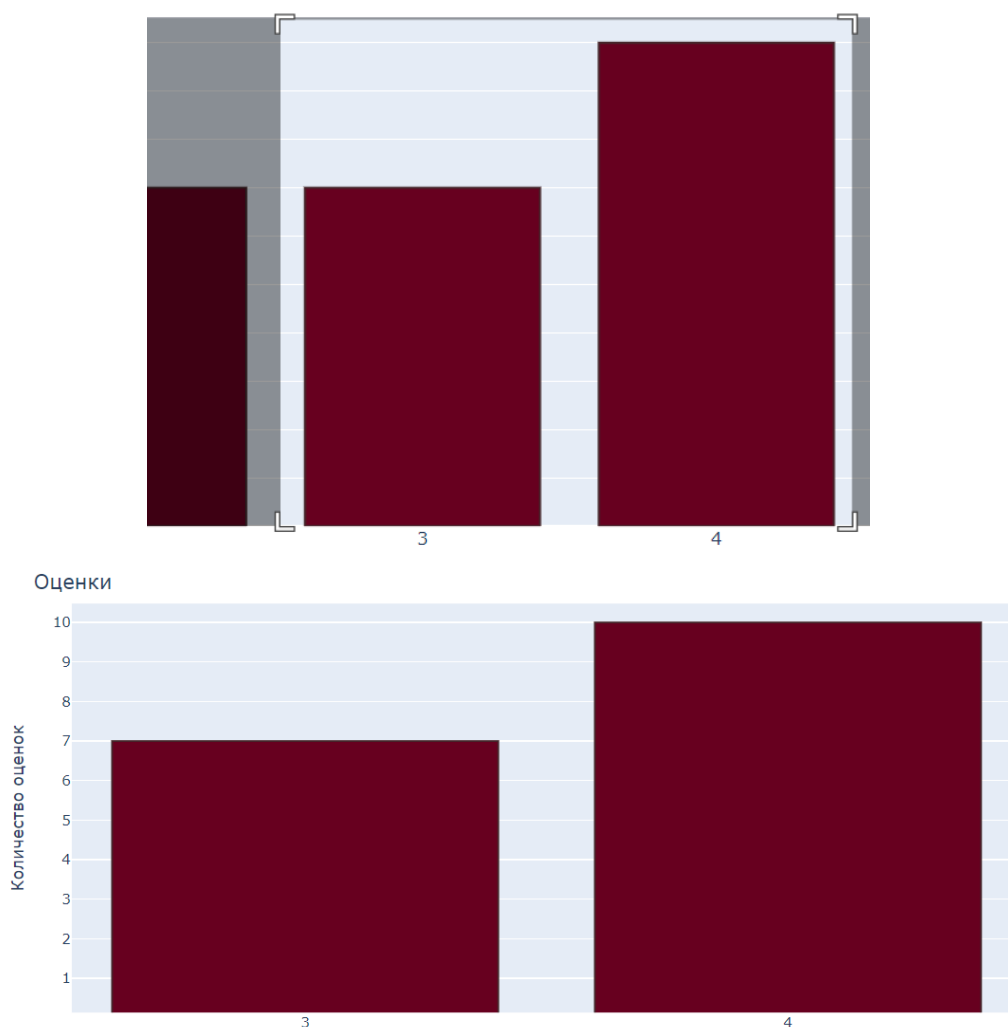


Рисунок 29 – Масштабирование графика

Также в функционале plotly присутствует панель, всплывающая у верхней границы поля с графиком, представляющая различные возможности в масштабировании графика, также там присутствует кнопка сохранения графика как изображения, что будет полезно при необходимости экспортирования визуального представления данных.

Пример описанной выше панели приведена на рисунке 30.

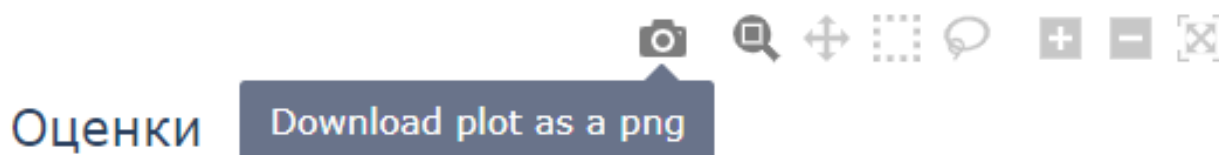


Рисунок 30 – Панель с дополнительными функциями

Чтобы перейти к созданной аналитической панели администратору из любого места системы была создана кнопка «Дашборд» в верхней панели системы.

Кнопка «Дашборд» представлена на рисунке 31.

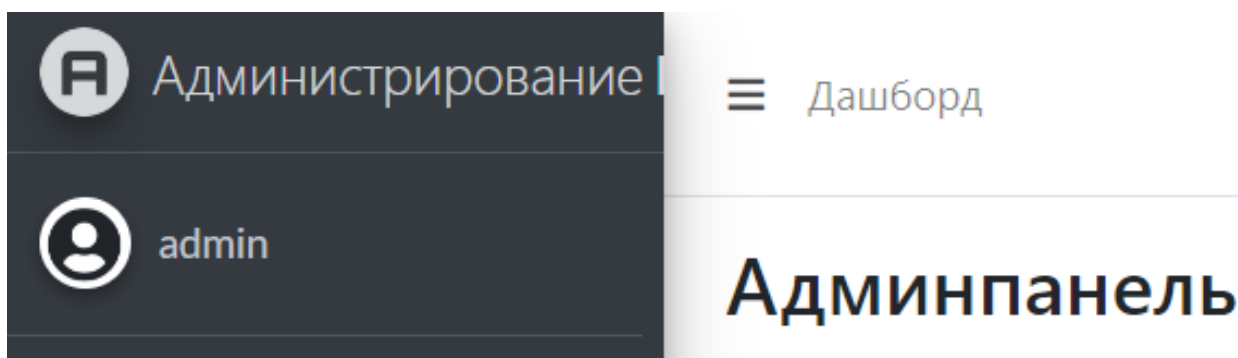


Рисунок 31 – Созданная кнопка перехода к аналитической панели

2.5 Выводы по главе 2

В итоге, для размещения в аналитической панели были выбраны визуализации:

- столбчатая диаграмма распределения оценок,
- кольцевая диаграмма, отражающая посещаемость студентов,
- таблица со студентами и их табельными номерами.

Для более детального отображения информации были реализованы фильтры данных по учебным группам и временным промежуткам. По учебным группам фильтр реализован в виде выпадающего меню, а по временным промежуткам фильтр реализован в виде календаря, на котором выбирается начальная и конечная даты промежутка времени.

Также в данной главе были описаны задачи по визуализации данных. В результате было сделано:

- были собраны данные с ведомостей с оценками обучающихся в технической школе за период с 2008 по 2019 годы;
- собранные данные были проанализированы и результатом анализа были выдвинуты требования к аналитической панели;
- по выдвинутым требованиям была построена аналитическая панель;
- построенная аналитическая панель интегрирована в ИИС;
- были получены навыки по интеграции диаграмм в проекты написанные на Django с помощью `django-plotly-dash`;
- были укреплены навыки в работе с `plotly express`, `Django`, `Dash`.

В качестве средства для визуализации был использован `django-plotly-dash`.

3 РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

3.1 Модуль экспорта отчетов

В процессе организации учебного процесса в технической школе необходимо формировать отчеты, поэтому была поставлена задача разработать модуль в ИИС для автоматического экспорта данных в полноценный документ для печати и дальнейшего редактирования.

Фрагмент программного кода подготовки данных для формирования файла `xlsx` представлен на рисунке 32.

```
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
filename = 'out.xlsx'
filepath = BASE_DIR + '/hours_app/download_files/' + filename
df = pd.DataFrame(data)
df['teacher'] = df['teacher'].astype(str)
df[['trash', 'teacher']] = df['teacher'].str.split(',', 1, expand=True)
df[['teacher', 'trash']] = df['teacher'].str.split(',', 1, expand=True)
df = df.drop(columns='trash')
df = df.rename(columns={"teacher": "Преподаватель", "sum_working": "в рабочее время", "sum_personal":
df.to_excel(filepath, encoding="utf-8")
```

Рисунок 32 – Фрагмент программного кода подготовки данных для формирования файла `xlsx`

Фрагмент кода с функцией скачивания файла представлен на рисунке 33.

```
def download_file(request):
    BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
    filename = 'out.xlsx'
    filepath = BASE_DIR + '/hours_app/download_files/' + filename
    path = open(filepath, 'rb')
    mime_type, _ = mimetypes.guess_type(filepath)
    response = HttpResponse(path, content_type=mime_type)
    response['Content-Disposition'] = "attachment; filename=%s" % filename
    return response
```

Рисунок 33 – Фрагмент программного кода с функцией скачивания файла

Во всех соответствующих ведомостях, находящихся в корпоративной системе, подключается модуль экспорта, функционал модуля экспорта заключается в преобразовании веб-страницы в полноценный документ для печати и дальнейшего редактирования.

На необходимую страницу добавляется кнопка экспорта, пример страницы представлен на рисунке 34.

Учёт часов (год)

Дата (будет использован год):

Учёт часов преподавания

преподаватель	итого за год	в личное время	в рабочее время
Преподаватель vasya, Иванов Иван Иванович, None	4	0	4
Преподаватель vlad, Смирнов Владислав Игоревич, None	6	0	6
Преподаватель andrey, Андреев Андрей Андреевич, None	8	0	8
Преподаватель evgeny, Жуков Евгений Романович, None	4	0	4

Рисунок 34 – Пример страницы с данными для экспорта и созданной кнопкой экспорта

После нажатия на кнопку экспорт пользователя перенаправляет на страницу настроек экспорта.

На рисунке 35 изображен интерфейс страницы модуля.

Формат:

- ☐ XLS
- ☐ DOCX
- ☐ PDF (не работает с кириллицей)

Имя документа:

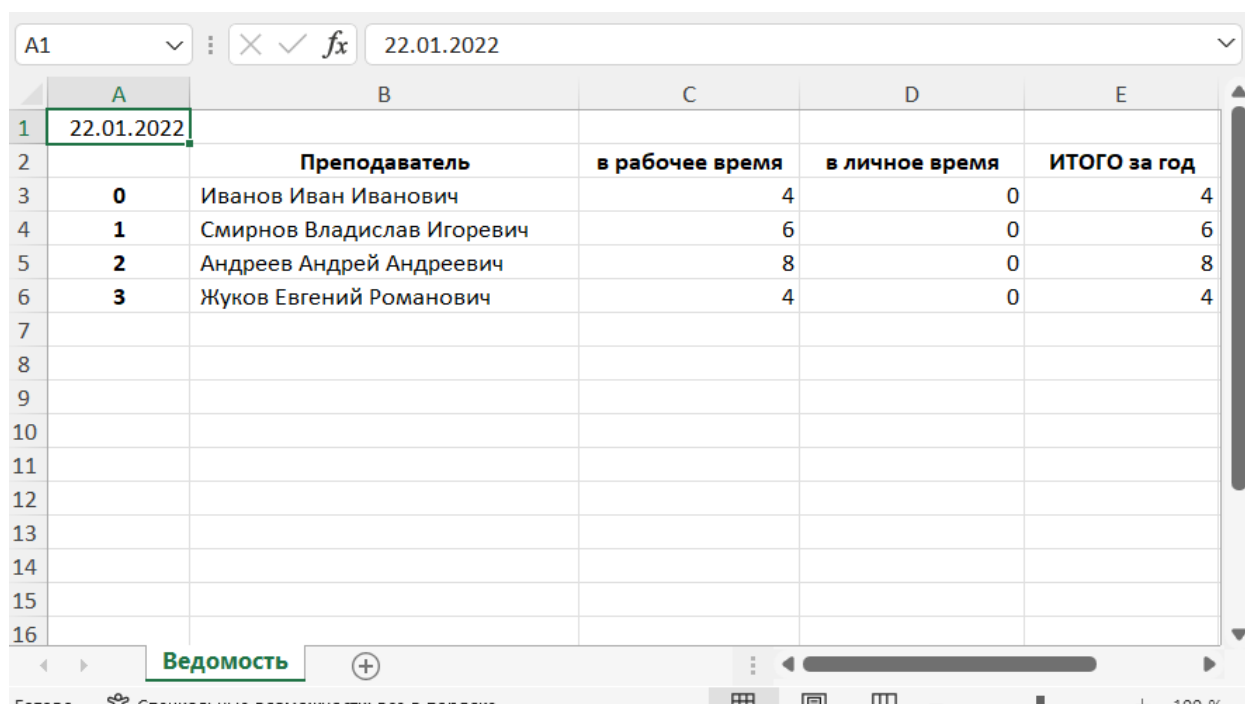
Дата:

Подпись: ☐

Рисунок 35 – Интерфейс модуля экспорта документов

Пользователь может выбрать формат, в котором нужно сохранить документ (xls, docx, pdf), задать название, отредактировать дату. В итоге, по нажатию кнопки «Экспорт» происходит сохранение документа в файловую систему.

Пример документа, полученного в результате работы модуля экспорта с выбранным форматом xls приведен на рисунке 36.

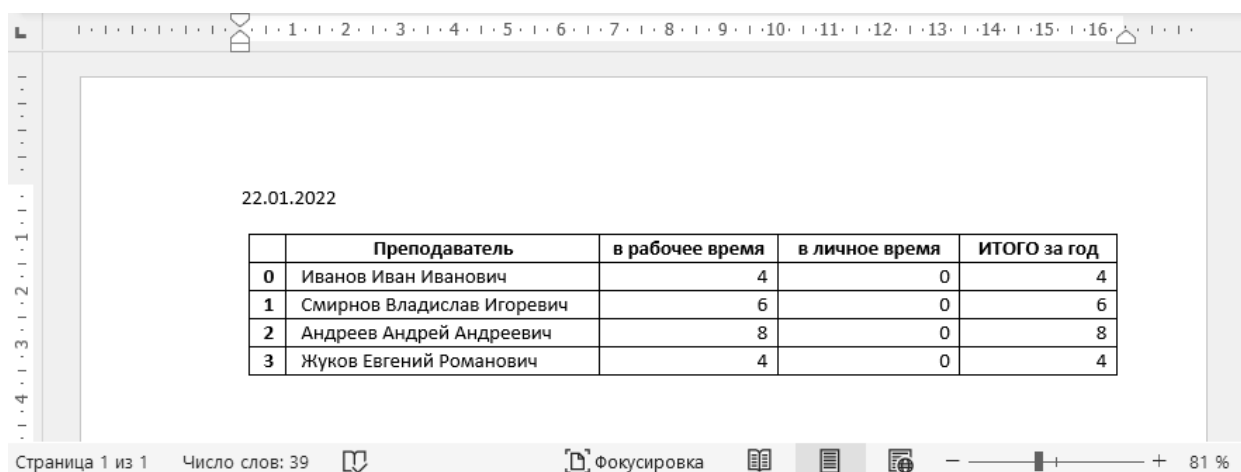


The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D	E
1	22.01.2022				
2		Преподаватель	в рабочее время	в личное время	ИТОГО за год
3	0	Иванов Иван Иванович	4	0	4
4	1	Смирнов Владислав Игоревич	6	0	6
5	2	Андреев Андрей Андреевич	8	0	8
6	3	Жуков Евгений Романович	4	0	4
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					

Рисунок 36 – Пример документа, полученного в результате работы модуля экспорта с выбранным форматом xls

Пример документа, полученного в результате работы модуля экспорта с выбранным форматом docx приведен на рисунке 37.



The screenshot shows a Word document with the following data:

22.01.2022

	Преподаватель	в рабочее время	в личное время	ИТОГО за год
0	Иванов Иван Иванович	4	0	4
1	Смирнов Владислав Игоревич	6	0	6
2	Андреев Андрей Андреевич	8	0	8
3	Жуков Евгений Романович	4	0	4

Рисунок 37 – Пример документа, полученного в результате работы модуля экспорта с выбранным форматом docx

3.2 Создание возможности авторизации студента

В предоставленной системе в качестве пользователей системы могли выступать преподаватели и администратор.

Передо мной была поставлена задача реализовать возможность пользоваться системой студентам, для просмотра оценок и посещаемости. На рисунке 38 приведена модель пользовательских сценариев, отражающая основные сценарии, уже присутствовавшие в системе и новые.



Рисунок 38 – Модель пользовательских сценариев

На рисунке 39 представлен фрагмент кода с моделью «Студент».

```
class Student(models.Model):
    """Модель "Студент". Объекты класса -- студенты ТШ."""
    user = models.ForeignKey(User, on_delete=models.CASCADE, verbose_name='Пол')
    personal_info = models.ForeignKey(PersonalInfo, on_delete=models.CASCADE,
    personnel_num = models.CharField('Табельный номер', max_length=6)
```

Рисунок 39 – Фрагмент кода с моделью «Студент»

Была создана группа пользователей «Студенты» (рисунок 40), и наделена необходимыми правами.

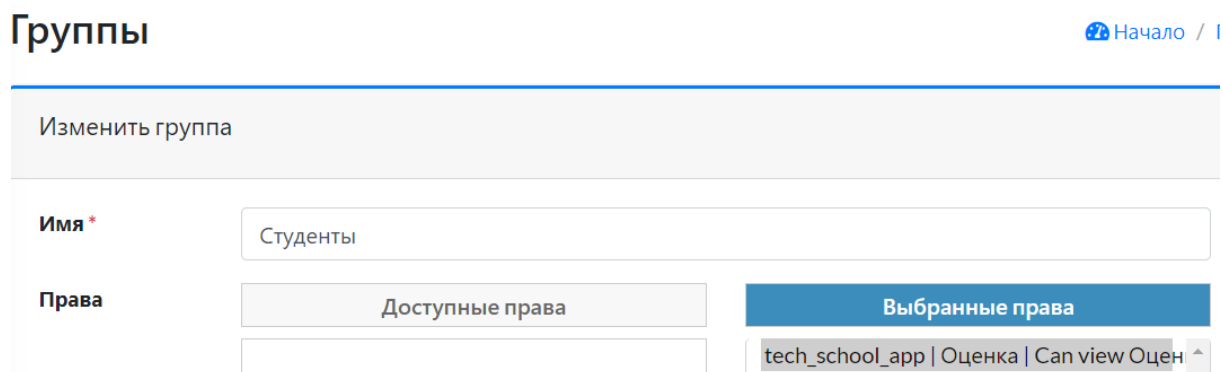


Рисунок 40 – Группа пользователей «Студенты»

3.3 Визуальное представление производственного календаря

В процессе построения учебного процесса необходимым элементом является производственный календарь. Он содержит информацию об общем количестве выходных, рабочих и праздничных дней в году.

В разрабатываемой ИИС для ТШ уже был реализован производственный календарь, но ввод в него данных был весьма затруднителен. Из-за большого объема данных ручной ввод их в систему практически невозможен, поэтому было решено провести автоматизацию их импорта.

Ввод дат производственного календаря осуществляется путем добавления каждой даты по отдельности, что крайне неудобно.

Интерфейс добавления даты производственного календаря в ИИС приведен на рисунке 41.

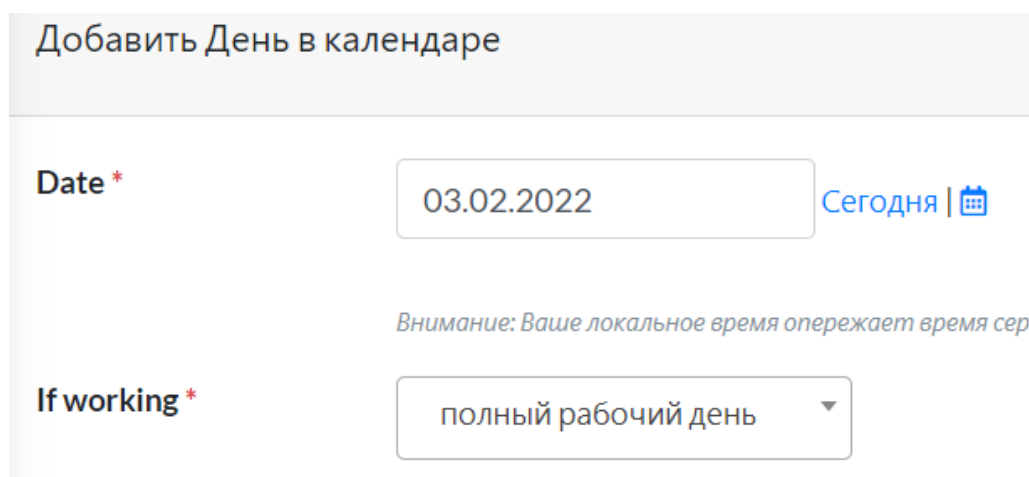


Рисунок 41 – Добавление даты производственного календаря в ИИС

Был написан скрипт, скачивающий с интернета производственный календарь в формате JSON и создающий CSV файл, необходимой структуры, для импорта в базу данных.

Скрипт – это небольшая программа, созданная для автоматического выполнения задачи и состоящая из последовательности действий.

На рисунке 42 изображен фрагмент кода, открывающий производственный календарь в формате JSON и создающий на основе него набор данных.

```
f = open('workersdays.json')
data = json.load(f)

holidays_list = []
if "holidays" in data:
    for i in data['holidays']:
        holidays_list.append(i)

preholidays_list = []
if "preholidays" in data:
    for i in data['preholidays']:
        preholidays_list.append(i)

nowork_list = []
if "nowork" in data:
    for i in data['nowork']:
        nowork_list.append(i)

if (len(holidays_list) > len(preholidays_list)):
    preholidays_list += (len(holidays_list)-len(preholidays_list)) * [nan]
if (len(holidays_list) < len(preholidays_list)):
    holidays_list += (len(preholidays_list)-len(holidays_list)) * [nan]
if (len(holidays_list) > len(nowork_list)):
    nowork_list += (len(holidays_list)-len(nowork_list)) * [nan]

workers_days = pd.DataFrame({'holidays': holidays_list,
                             'preholidays': preholidays_list,
                             'nowork': nowork_list})
```

Рисунок 42 – Фрагмент кода скрипта, создающий набор данных производственного календаря

На рисунке 43 приведен пример полученного набора данных.

	holidays	preholidays	nowork
0	2022-01-01	2022-02-22	NaN
1	2022-01-02	2022-03-05	NaN
2	2022-01-03	2022-11-03	NaN
3	2022-01-04	NaN	NaN
4	2022-01-05	NaN	NaN
...
113	2022-12-17	NaN	NaN
114	2022-12-18	NaN	NaN
115	2022-12-24	NaN	NaN
116	2022-12-25	NaN	NaN
117	2022-12-31	NaN	NaN

Рисунок 43 – Полученный набор данных производственного календаря

На рисунке 44 изображен фрагмент кода, создающий набор данных структуры, аналогичной структуре хранения производственного календаря в базе данных и заполняющий ее данными.

```

dates_ = pd.date_range(start='1/1/2022', end='31/12/2022')
id_ = list(range(1, len(dates_)+1))
df_structure = pd.DataFrame({'id': id_, 'date': dates_, 'if_working': nan, 'in_hours': nan})

df_structure['date'] = df_structure['date'].dt.date

for i in range(df_structure.shape[0]):
    if any(workers_days['preholidays'] == df_structure['date'][i]):
        df_structure['if_working'][i] = "7"
        df_structure['in_hours'][i] = 7
    elif any(workers_days['holidays'] == df_structure['date'][i]):
        df_structure['if_working'][i] = "np"
        df_structure['in_hours'][i] = 0
    elif any(workers_days['nowork'] == df_structure['date'][i]):
        df_structure['if_working'][i] = "np"
        df_structure['in_hours'][i] = 0
    else:
        df_structure['if_working'][i] = "8"
        df_structure['in_hours'][i] = 8

df_structure['in_hours'] = df_structure['in_hours'].astype(int)
df_structure['if_working'] = df_structure['if_working']

df_structure.to_csv("workersdays2022.csv", index=False, encoding="utf-8")

```

Рисунок 44 – Фрагмент кода, создающий набор с данными производственного календаря необходимой структуры

На рисунке 45 приведен пример полученного набора данных.

	id	date	if_working	in_hours
0	1	2022-01-01	Пр	0
1	2	2022-01-02	Пр	0
2	3	2022-01-03	Пр	0
3	4	2022-01-04	Пр	0
4	5	2022-01-05	Пр	0
...
360	361	2022-12-27	8	8
361	362	2022-12-28	8	8
362	363	2022-12-29	8	8
363	364	2022-12-30	8	8
364	365	2022-12-31	Пр	0

Рисунок 45 – Полученный набор данных производственного календаря, необходимой структуры

В результате работы описанного выше скрипта создается CSV файл, который можно импортировать в базу данных без особых усилий.

Поля, содержащиеся в полученном csv файле:

- 1) «date» – дата;
- 2) «if_working» – Маркер, по которому определяется статус дня:
 - «Пр» – праздник;
 - «8» – полный рабочий день;
 - «7» – короткий рабочий день;
- 3) «in_hours» – количество рабочих часов.

Полученный файл можно легко импортировать в базу данных благодаря тому, что его структура совпадает со структурой таблицы в базе данных.

Пример импортирования файла csv в базу данных, с помощью программы «pgAdmin 4» представлен на рисунке 46.

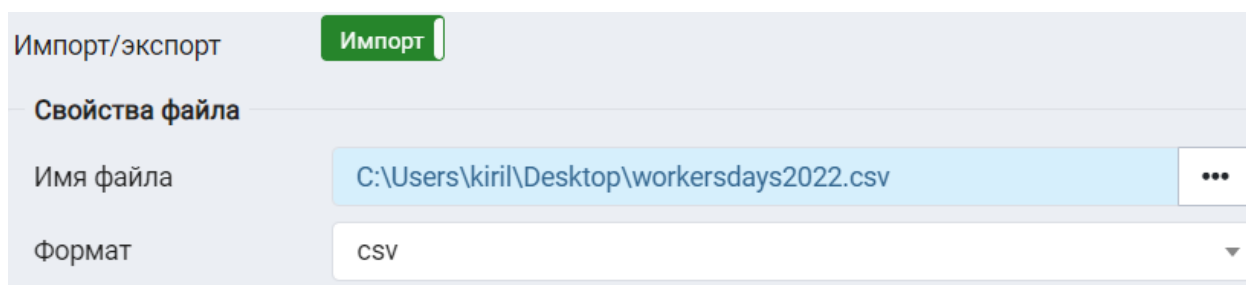


Рисунок 46 – Импортирование csv файла в базу данных

Для возможности просмотра в ИИС производственного календаря было создано его визуальное представление.

Созданное визуальное представление производственного календаря представлено на рисунке 47.

Январь	1 Праздник	2 Праздник	3 Праздник	4 Праздник	5 Праздник
Февраль	1 Полный рабочий день	2 Полный рабочий день	3 Полный рабочий день	4 Полный рабочий день	5 Праздник
Март	1 Полный рабочий день	2 Полный рабочий день	3 Полный рабочий день	4 Полный рабочий день	5 Короткий рабочий день
Апрель	1 Полный рабочий день	2 Праздник	3 Праздник	4 Полный рабочий день	5 Полный рабочий день
Май	1 Праздник	2 Праздник	3 Праздник	4 Полный рабочий день	5 Полный рабочий день
Июнь	1 Полный рабочий день	2 Полный рабочий день	3 Полный рабочий день	4 Праздник	5 Праздник
	— Полный	—	—	— Полный	— Полный

Рисунок 47 – Визуальное представление производственного календаря

Синим показаны полные рабочие дни, красным – праздничные дни, а желтым – короткие рабочие дни.

3.4 Выводы по главе 3

Был написан модуль для отчета в виде полноценного документа из данных системы.

Был создан пользователь студент и наделен правами по просмотру оценок и посещений.

Был написан скрипт для скачивания из интернета данных производственного календаря и формирования CSV файла для дальнейшей загрузки в систему.

Было создано визуальное представление производственного календаря.

ЗАКЛЮЧЕНИЕ

В рамках выпускной квалификационной работы было проведено функциональное тестирование полученной от подразделения «Техническая школа» ГУП «Петербургский метрополитен» системы, которая в настоящий момент разрабатывается. Тестирование было проведено в рамках определенного пользовательского сценария, затрагивающего наибольшее количество модулей системы. Выбранный сценарий – ведение электронного журнала. В результате проведенного тестирования была выявлена ошибка в создании занятия.

Выявленная ошибка была исправлена, а также была проведена локализация некоторых элементов системы.

Были собраны данные с ведомостей с оценками обучающихся в технической школе за период с 2008 по 2019 годы в формате JPEG. Собранные данные были перенесены в машиночитаемый формат CSV.

Был проведен анализ структуры данных в системе. В результате проведенного анализа структуры данных в системе были выдвинуты требования к аналитической панели.

По выдвинутым требованиям была спроектирована и построена интерактивная аналитическая панель для анализа успеваемости учебных групп и посещаемости студентами занятий, содержащая в себе столбчатую диаграмму распределения оценок и кольцевую диаграмму, отражающую посещаемость студентов, а также таблицу со студентами и их табельными номерами. Созданная аналитическая панель интерактивна, за счет наличия фильтрации данных по учебным группам и временным промежуткам.

Созданная аналитическая панель была интегрирована в разрабатываемую интеллектуальную информационную систему.

Был написан модуль для экспорта отчета в виде полноценного документа для печати и дальнейшего редактирования, который формируется из данных в системе.

Написанный модуль был интегрирован в систему.

Был создан пользователь студент и наделен правами по просмотру оценок и посещений.

Было создано визуальное представление производственного календаря. Был написан скрипт для автоматизации ввода данных производственного календаря в систему.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. He B. X., Zhuang K. J. Research on the intelligent information system for the multimedia teaching equipment management // 2016 International Conference on Information System and Artificial Intelligence (ISAI). – IEEE, 2016. – С. 129-132.
2. Guzmán-Valenzuela C. et al. Learning analytics in higher education: a preponderance of analytics but very little learning? // International Journal of Educational Technology in Higher Education. – 2021. – Т. 18. – №. 1. – С. 1-19.
3. Yandex DataLens // Yandex Cloud URL: <https://cloud.yandex.ru/services/datalens> (дата обращения: 13.04.2022).
4. Tableau Desktop // Tableau URL: <https://www.tableau.com/products/desktop> (дата обращения: 13.04.2022).
5. Power BI Desktop // Microsoft URL: <https://powerbi.microsoft.com/ru-ru/desktop/> (дата обращения: 13.04.2022).
6. Qlik Sense | Modern Cloud Analytics // Qlik URL: <https://www.qlik.com/us/products/qlik-sense> (дата обращения: 13.04.2022).
7. django-plotly-dash documentation // URL: <https://django-plotly-dash.readthedocs.io/en/latest/index.html> (дата обращения: 16.04.2022).
8. Dash Python User Guide // Plotly URL: <https://dash.plotly.com/> (дата обращения: 17.04.2022).
9. Django documentation // Django URL: <https://docs.djangoproject.com/en/4.0/> (дата обращения: 18.04.2022).
10. Plotly Express in Python // Plotly URL: <https://plotly.com/python/plotly-express/> (дата обращения: 19.04.2022).
11. Dropdown // Plotly URL: <https://dash.plotly.com/dash-core-components/dropdown> (дата обращения: 20.04.2022).
12. DatePickerRange // Plotly URL: <https://dash.plotly.com/dash-core-components/datepickerrange> (дата обращения: 20.04.2022).
13. Dash DataTable // Plotly URL: <https://dash.plotly.com/datatable> (дата обращения: 24.04.2022).