

Барнаул 2020

## Реферат

Данная работа описывает проектирование и разработку системы для расчета внутренней и внешней совместимости учебных материалов, её пользовательский интерфейс, процесс визуализации модели данных, сценарии действий пользователей системы, а также тестирование системы.

Объем работы – 51 страница, включающих 39 рисунков, 5 таблиц, 25 используемых источников.

Ключевые слова: предметный тезаурус, внутренняя и внешняя совместимость, учебные материалы, веб-приложение.

## The abstract

This paper describes the design and development of a system for calculating the internal and external compatibility of educational materials, its user interface, the process of visualizing the data model, scenarios for the actions of system users, and testing the system.

Size of work – 57 pages includes 39 figures, 5 tables, 25 used sources.

Keywords: the subject thesaurus, internal and external compatibility, training materials, web-based application.

					<i>БР 09.03.04.05.000 ПЗ</i>			
<i>Изм</i>	<i>Лист</i>	<i>№ докум</i>	<i>Подп.</i>	<i>Дата</i>	<i>Разработка web-интерфейса для анализа учебных материалов с учетом их внутренней и внешней совместимости</i>	<i>Лит</i>	<i>Лист</i>	<i>Листов</i>
<i>Разр.</i>		<i>Оверченко В.Р.</i>				<i>У</i>	<i>2</i>	<i>57</i>
<i>Руководитель</i>		<i>Андреева А.Ю.</i>				<i>АлмГТУ      ФИТ ПИ - 61</i>		
<i>Н. контр</i>		<i>Потупчик А. И.</i>						
<i>Зав. каф.</i>		<i>Кантор С. А.</i>						

# Содержание

Введение	4
1 Описание предметной области	5
1.1 Основные определения	5
1.2 Современные подходы к анализу	6
1.3 Обзор существующих решений	11
1.3.1 Программный продукт “Анализ учебной письменной речи”	11
1.3.2 Программный продукт “LightReader”	15
1.3.3 Программный продукт “Панда-копирайтинг”	16
1.3.4 Программный продукт “Miratext - биржа копирайтеров”	19
1.4 Функциональные возможности разрабатываемого приложения	21
2 Проектирование web-приложения	23
2.1 Структура приложения	23
2.2 Описание процессов взаимодействия пользователей и системы	24
2.2.1 Описание групп пользователей	24
2.2.2 Описание процесса регистрации и авторизации	25
2.2.3 Описание процесса работы с одним учебным материалом	25
2.2.4 Описание процесса работы с двумя учебными материалами	26
2.2.4 Описание процесса работы с личным кабинетом	26
3 Выбор инструментов разработки	28
3.1 Клиентская часть	28
3.2 Серверная часть	28
3.3 Дополнительные инструменты	29
3.4 Библиотека для расчёта совместимости учебных материалов	30
4 Архитектура приложения	31
4.1 База данных	31
4.2 Описание архитектуры веб-приложения	33
4.3 Интерфейс приложения	35
5 Тестирование	41
Заключение	46
Список использованных источников	47
Приложение А Задание на выполнение ВКР	50
Приложение Б Исходный код программы	52

## **Введение**

Многие преподаватели и учителя часто задаются вопросом, какими знаниями должен обладать студент или школьник на данный момент времени по результатам изучения смежных дисциплин и предыдущих курсов. Знакомы обучающемуся те или иные понятия, теоремы, формулы, явления, процессы. Не найдется, пожалуй, ни одной учебной дисциплины, которая не имеет связи с другой. Так, например, неразрывно связаны между собой экономика Алтайского края и география. Для полного понимания особенностей экономики тяжелой промышленности следует знать о месторождениях полезных ископаемых. Таких примером можно приводить очень много. И это касается не только разных курсов. Даже в рамках одного предмета очень важна логически построенная структура информации.

В связи с этим для работников образовательных учреждений возникает соответствующая задача по анализу учебных материалов на совместимость. Ручной анализ может занять очень много времени и не всегда быть полным и достоверным. Именно поэтому появляется потребность в автоматизации данного процесса с предоставлением удобного инструмента для анализа преподавателям, учителям и другим сотрудникам образовательной и научной сферы. Так как необходимость в подобном анализе возникает практически у каждого работника, то целесообразно предоставить соответствующий инструмент на web-платформе.

Целью данной работы является разработка удобного и функционального web-интерфейса для сокращения трудовых затрат работников образования, связанных с анализом учебных материалов на совместимость.

Для исследования были поставлены следующие задачи:

- 1) Провести анализ существующих аналогов;
- 2) Спроектировать сценарий анализа учебных материалов на внутреннюю и внешнюю совместимость;
- 3) Разработать архитектуру приложения и базу данных;
- 4) Реализовать web-приложение, являющееся оболочкой над модулем расчёта совместимости;
- 5) Протестировать реализованное приложение.

# 1 Описание предметной области

## 1.1 Основные определения

Система учебных текстов – конечное множество символов, знакообъединений, словоформ, словосочетаний, предложений и их групп, объединенных динамическими и статическими отношениями, которые с необходимостью и достаточностью обуславливают наличие целенаправленного свойства – текстовой совместимости, позволяющей диалектически решать многомерное дихотомическое противоречие между связностью и обособленностью письменной речи на выбранном комплекте вузовских учебников.

Совместимость учебно-научных текстов – междисциплинарная категория, отражающая диалектический тип отношений между связностью и обособленностью, а также и всеми выделенными дополнительными противоречиями, проявляющаяся при максимальной дихотомической мерности сложности, обусловленная задачами высшего образования.

Другими словами, рассмотрение категории “совместимости вообще” должно осуществляться не как изолированной категории, а в рамках дихотомии *связное - обособленное* (Рисунок 1).



Рисунок 1 - Дихотомия совместимости

Можно выделить совместимость учебных материалов двух типов (Рисунок 2).



Рисунок 2 - Типы совместимости учебных материалов

Внутренняя совместимость отражает совместимость в пределах одного учебного текста, например, между главами или разделами. Внешняя совместимость - между материалами различных дисциплин.

Мы хотим представить совместимость учебных материалов как некоторую количественную величину, измеряемую в процентах.

Считается, что для внутренней совместимости прирост предметного тезауруса должен составлять не более 20%.

Для полного понимания исследуемой области стоит также выделить понятие предметного тезауруса.

Предметный тезаурус — словарь, собрание сведений, корпус или свод, полномерно охватывающие понятия, определения и термины специальной области знаний или сферы деятельности, что должно способствовать правильной лексической, корпоративной коммуникации.

## 1.2 Современные подходы к анализу

Компьютерное моделирование предоставляет большие возможности для исследований не только в точных науках, но и в некоторых гуманитарных исследованиях, где параметры исследуемых объектов могут быть трудно формализованы. Методы моделирования становятся мощным инструментом в

различных областях знаний, в том числе в педагогике, дидактике<sup>1</sup> и смежных областях.

Мы исследовали такой значимый элемент образовательного процесса, как учебная информация и способы ее передачи, отражающие бессознательный спектр ее восприятия и усвоения. Таким образом показатели образовательной информации выступают объектом моделирования. Анализ научных публикаций показал противоречивость подходов, используемых для выбора аналитических показателей и методов моделирования. Это можно объяснить спецификой и многомерной сложностью формализованного объекта.

Для создания модели анализа системы учебных текстов на предмет их совместимости друг с другом необходимо обеспечить согласованность последовательных уровней сложности учебного материала для его последующего преобразования в знания. В работах Рыбаковой Г.Р.<sup>2</sup> и Клочкова В.П.<sup>3</sup> предлагается достичь этой цели путем последовательного обогащения тезауруса. В качестве базового понятия их исследования является фоносемантическое поле, которое понимается как система со связями между ее взаимодействующими элементами— знаковыми структурами и семантическим содержанием. Эти связи трудно формализовать, поскольку они затрагивают семантический аспект передаваемого знания.

При анализе семантической совместимости учебной информации и ее влияния на бессознательные процессы усвоения и запоминания учебного материала следует учитывать его полевую природу. Фоносемантическое пространство представляет собой единую структуру семантических и символических единиц, сгруппированных вокруг семантического ядра (Рисунок 3). Сила связи между ними оценивается путем

---

<sup>1</sup> Раздел педагогики, излагающий общую теорию образования и обучения.

<sup>2</sup> Рыбакова Галина Раисовна - кандидат биологических наук, доцент Сибирского федерального университета Торговоэкономического института (Красноярск).

<sup>3</sup> Клочков Владимир Павлович - вице-президент Международной академии фундаментального образования, руководитель её Сибирского отделения; академик Нью-Йоркской академии наук; член.-корр. Международной академии психологических наук; член.-корр. Российской академии естествознания (РАЕ); заслуженный деятель науки и образования РАЕ; доктор педагогических наук, доктор философии (PhD), доктор психологии (Psy.D), доктор делового администрирования (DBA); профессор кафедры психологии и управления.

сравнения частотности ключевых слов (семантического ядра) и других слов, используемых с ядром и сгруппированных вокруг него. Сила внутренних связей уменьшается от семантического центра к периферии поля, но появляются новые связи, внешние для данного поля: это связи с другими семантическими полями и их пересечениями. Таким образом, в рамках системного подхода фоносемантическое поле можно рассматривать как совокупность элементов со связями и отношениями между ними.



Рисунок 3 - Фоносемантическое поле

Для определения силы связи между семантическими и символическими единицами можно воспользоваться статической мерой TF-IDF (от англ. TF — term frequency, IDF — inverse document frequency). Данная мера используется для определения важности слова в контексте документа. TF-IDF - произведение TF-меры (отношения числа вхождений слова к общему числу слов документа) и инверсии частоты встречаемости слова в документах корпуса (IDF).

TF-мера определяет важность слова  $t$  в пределах отдельного документа  $d$  и определяется как

$$tf(t, d) = \frac{n_t}{\sum_k n_k},$$

где  $n_t$  - это число вхождений слова в документе, а в знаменателе сумма вхождений всех слов в документе.



IDF – обратная частота документа, с которой слово встречается в документах коллекции.

$$\text{idf}(t, D) = \log \frac{|D|}{|\{d_i \in D \mid t \in d_i\}|}$$

где  $|D|$  – число документов коллекции,

$|\{d_i \in D \mid t \in d_i\}|$  – число документов из коллекции  $D$ , в которых встречается слово  $t$ .

Выбор основания логарифма не важен, так как значение изменится у всех на определенный множитель, что никак не повлияет на соотношения весов.

Таким образом, мера TF-IDF вычисляется путем умножения TF на IDF.

$$\text{tf-idf}(t, d, D) = \text{tf}(t, d) \times \text{idf}(t, D)$$

Стоит отметить, чем больше частота использования слова в текстах, тем ниже будет значение меры IDF (стремится к нулю). Это относится не только к общей лексике (глаголы-связки, служебные части речи), но и к терминам, преобладающим в документе. Одновременно, общая лексика, обозначающая одни и те же ситуации либо действия с разных точек зрения и задающие так называемые конверсивные замены («приводить  $\Leftrightarrow$  являться следствием»), будут иметь более высокие значения меры IDF. Можно предположить, что наиболее уникальные слова в документе (с наибольшими значениями произведения мер TF и IDF) будут относиться к терминам предметной области документа. Если слово имеет синонимы, упомянутые в этом же документе, значение меры TF для этого слова будет ниже. Как и в случае конверсивных замен, имеем меньшую встречаемость в документах корпуса каждого слова из синонимического ряда и, следовательно, более высокие значения меры IDF по сравнению со случаем отсутствия синонимов у слова.

Следует отметить, что применимость такого подхода к анализу ключевых слов определяется спецификой учебной информации. Эта информация написана в научном стиле, и ее смысл существенно отличается от других видов текстовой информации. Терминологическая плотность — это условие однозначной интерпретации содержания. Кроме того, использование специальной терминологии

приводит к высокому сгущению смысла в словах и словосочетаниях, а также значительно увеличивает частоту их употребления

Для того чтобы образовательная информация была адекватно декодирована, необходимо выполнить определенные требования: логика изложенного материала должна соответствовать логике мышления, стоящего за ним, а также логике науки, к которой относится передаваемая информация. Между различными частями учебного текста должны быть логические связи. На уровне учебника такие связи определяются содержанием текста, а на уровне курса или университетской специальности-учебным планом.

Компьютерный анализ текстов основан на том же логическом принципе. Условием формализации является логическое композиционное структурирование информационного массива, а результатом-возможность применения математических операций обработки данных к семантическому содержанию текстов.

Логика сознательного восприятия может быть проанализирована путем построения графа логических связей, вершинами которого будут являться смысловые центры (ключевые слова), с соответствующей матрицей логических связей. Использование аналогичной методики, апробированной И. Б. Моргуновым (2007) для изучения последовательности учебных дисциплин в учебном плане, позволит структурировать учебную информацию путем определения потенциального числа семантических ядер и оценки функции общего числа разрывов логических связей для конкретных текстовых блоков в рамках дисциплин каждого цикла. Это помогает определить оптимальную последовательность изучения материала при минимальном количестве нарушенных логических связей.

Таким образом, приходим к следующему умозаключению - для расчёта совместимости учебных материалов можно опираться на такой алгоритм:

- 1) Выделение предметного тезауруса ключевых слов для каждого анализируемого учебного материала;
- 2) Применение к тезаурусам статической меры TF-IDF;
- 3) Вычисление процентной совместимости материалов на основе полученных данных по формуле:

Совместимость = Количество совпадающих слов предметного тезауруса в первом и втором текстах / Количество слов предметного тезауруса во втором тексте \* 100%.

### **1.3 Обзор существующих решений**

#### **1.3.1 Программный продукт “Анализ учебной письменной речи”**

Группой авторов А.Д.Никина, Н.К.Криони и А.В.Филипповой. на основе методов математического программирования была разработана информационная система «анализа учебной письменной речи». Они автоматизировали процесс анализа учебного материала на сложность восприятия с дальнейшей адаптацией обучающимися.

В своей работе они придерживались следующих идей:

- Проблема трудности восприятия учебного текста может рассматриваться в двух аспектах: в аспекте генерации текста и в аспекте восприятия текста;
- Имея объективную оценку сложности учебного текста, преподаватель может оценить время, необходимое студенту на изучение текста, а студент, используя указания на элементы текста, определяющие его сложность, сможет при его изучении обратить особое внимание на данные фрагменты текста;
- Перечень ключевых слов, будучи предъявлен читателю перед чтением текста, способствует лучшему усвоению текста.

В связи с этим созданный программный продукт отвечал следующим функциям:

- автоматическое выделение ключевых слов;
- автоматический поиск параметров сложности текстов и их подсчет;
- диалог с пользователем с целью проверки правильности выделенных параметров;
- сохранение результатов анализа сложности текста и поиска в нем ключевых слов в отдельных файлах;

- возможность работы с документами в форматах doc, txt, rtf.

Обобщенная структура программы представлена на рисунке 4.

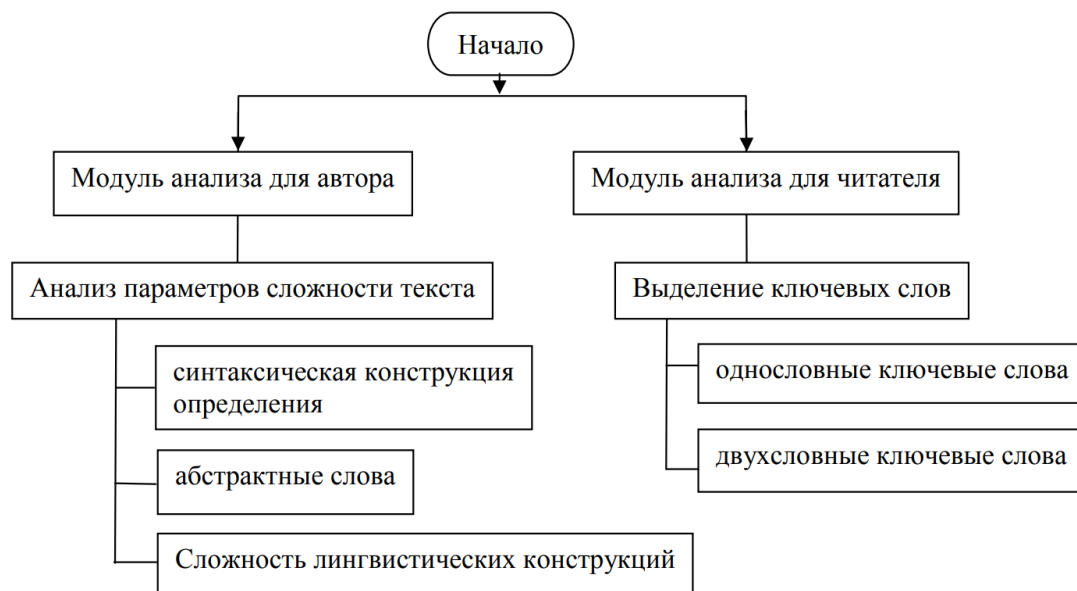


Рисунок 4 - Обобщённая структура программы “Анализ сложности учебных текстов”

В качестве параметров сложности текста выступали:

- синтаксическая конструкция определения;
- абстрактные слова;
- сложность лингвистических конструкций.

Алгоритм работы автора текста представлен на рисунке 5.

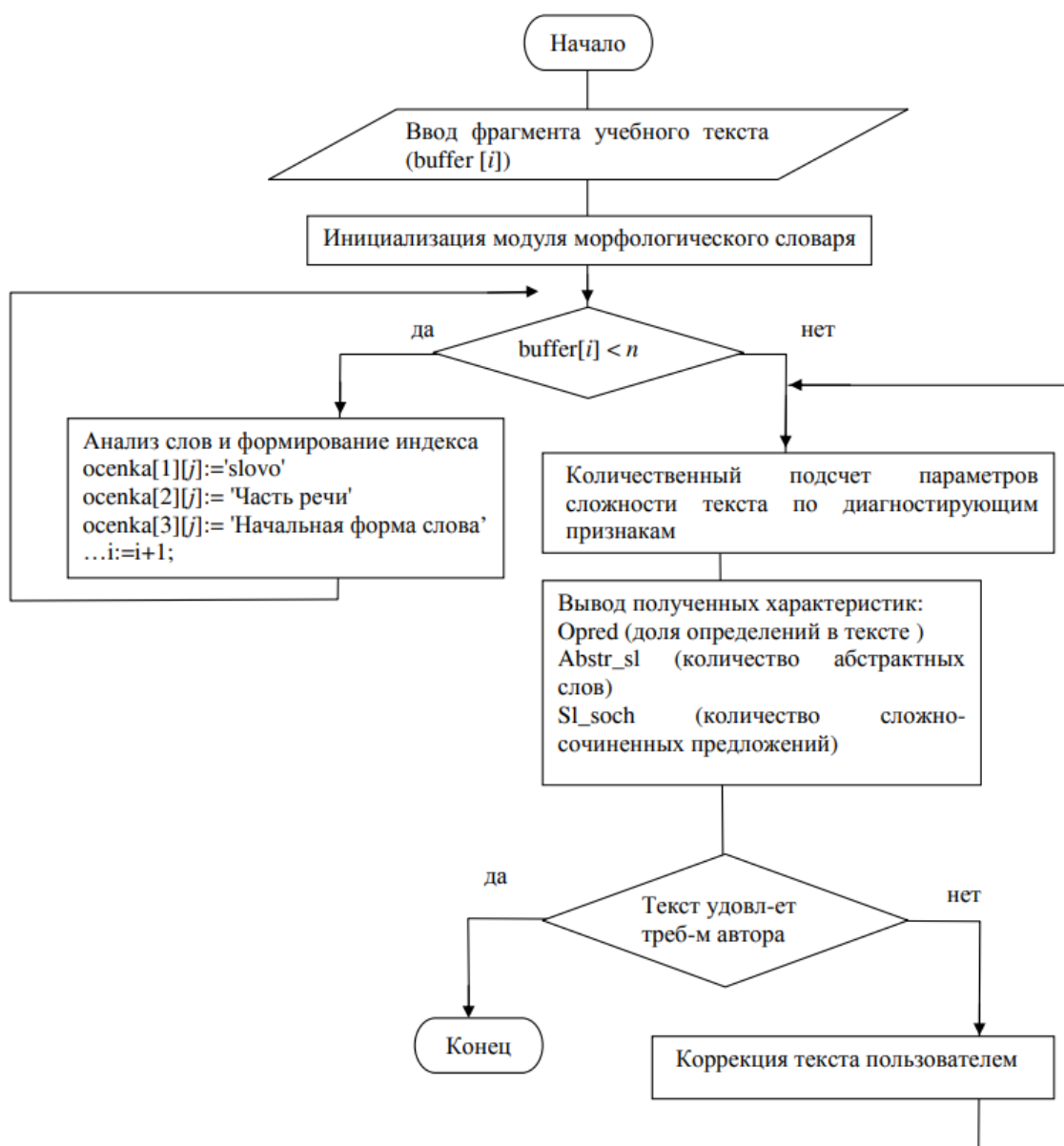


Рисунок 5 - Структура модуля анализа для автора

В качестве инструментального средства проектирования (пакета прикладных программ) был выбран продукт Borland Delphi 6.0. Для работы с естественным языком были использован морфологический словарь, содержащий все словоформы русского языка. В программе анализ наличия параметров сложности в тексте и их поиск производился на основе данных о словоформе каждого отдельного слова из текста. В качестве метода представления и сбора данных о слове был выбран метод индекса, когда для каждого термина записывается набор страниц, где может этот термин находиться. Аналогичный принцип использовался для работы программы, был создан многомерный массив, хранящий набор информации о слове: исходную

форму слова, морфологическую часть речи, словоформу, количество гласных, номер нахождения в тексте. Анализ многомерного массива позволяет выявить наличие параметров сложности в тексте, выполнить их подсчет и вывести данные на экран. По результатам первичного анализа выводится таблица количественных характеристик параметров сложности текста, далее автору предлагается отдельно поработать с каждым параметром текста, а именно — просмотреть каждую группу найденных параметров сложности текста. Автор текста может либо оставить в тексте все как есть, либо упростить конструкцию предложения, убрать абстрактное слово, изменить определение и провести повторный анализ.

Интерфейс программы для автора представлен на рисунке 6.

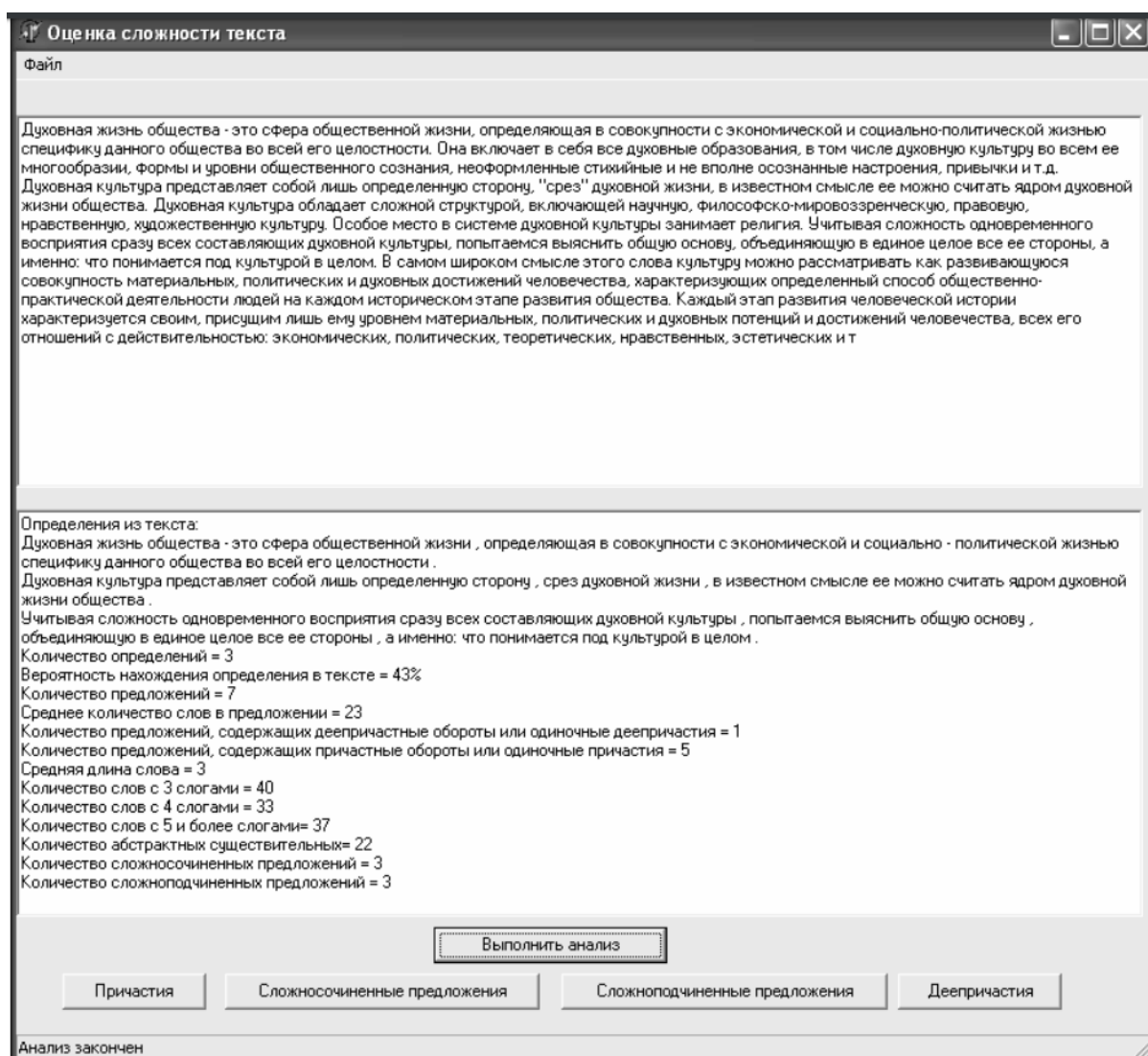


Рисунок 6 - Окно модуля анализа для автора

### 1.3.2 Программный продукт “LightReader”

Схожая по функционалу программа LightReader представлена в работе И.В.Оборневой “Автоматизированная оценка сложности учебных текстов на основе статических параметров”. Автором с использованием методов математического программирования была разработана методика автоматизированной оценки сложности учебных текстов. При этом использовались только статистические показатели, такие как среднее число слов в предложении, среднее число слогов в слове, число многосложных слов и другие.

Программный продукт LightReader являлся макросом для программного продукта Microsoft Office Word. Программа состоит из двух модулей: модуль для расчёта статических показателей текста и оценки его сложности, и модуль анализа текста в зависимости с заданными условиями относительно возраста учащихся.

Программа может обрабатывать текст не только на русском, но и на английском языке. В результате обработки получаем таблицу значений статических показателей и оценку сложности текста, рассчитанную по формуле Флеша:

$$K = 206,836 - 60,1 W - 1,3 - S, \text{ где}$$

$K$  - оценка трудности текста от 0 до 100,

$W$  - среднее число слогов в слове,

$S$  - среднее количество слов в предложении.

Дизайн продукта соответствует стандартному стилю Windows-приложений. Для демонстрации работоспособности программы использована последняя версия Microsoft Office Word (рисунок 7, рисунок 8).

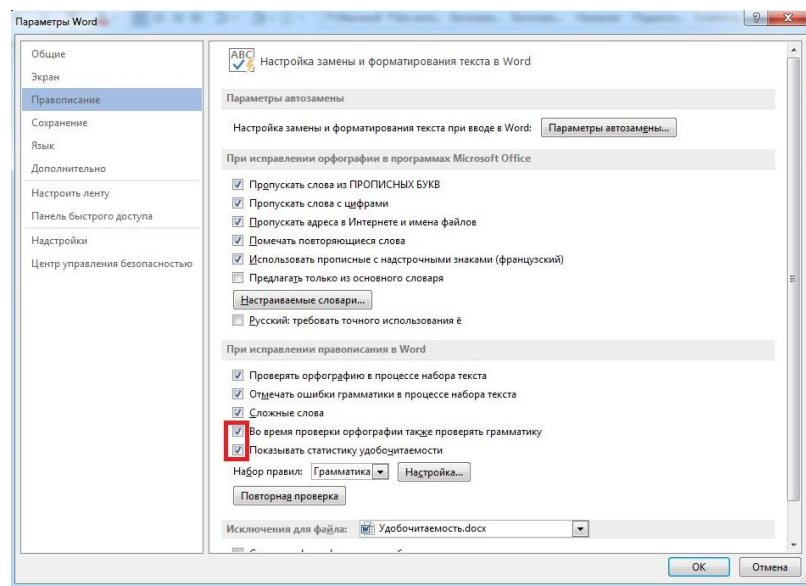


Рисунок 7 - Подключение статистики удобочитаемости

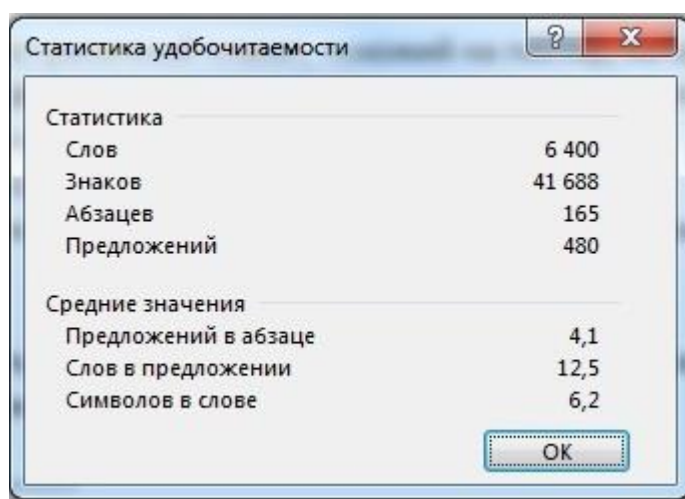


Рисунок 8 - Результат анализа текста на удобочитаемость

### 1.3.3 Программный продукт “Панда-копирайтинг”

Панда-копирайтинг представляет собой веб-приложение для работы с текстом. Обладает широким функционалом, начиная от генератора заголовков для лендингов 4U до поиска тавтологий “Свежий взгляд”. особый интерес для нашей работы представляет раздел “Сравнение 2 текстов на схожесть” - онлайн сервис для поиска отличий.

Работа сервиса имеет несколько особенностей:

- для анализа можно добавить тексты вручную либо загрузить файлами;
- после загрузки тексты хранятся в буфере обмена 15 минут, а затем буфер очищается;



- тексты можно сравнивать как в общем массиве (изменения выделяются цветом), так и отдельно в колонках.

Интерфейс страницы для работы с текстом представлен на рисунке 9.

Алгоритм работы с сервисом следующий:

- 1) Загрузка файла / набор текста вручную для анализа в левой колонке.  
Здесь предполагается загрузка оригинального текста (до изменений);
- 2) Загрузка файла / набор текста вручную для анализа в правой колонке.  
Здесь должен быть текст с поправками или изменениями;
- 3) Запуск процесса для сравнения текстов по нажатию на соответствующую кнопку.

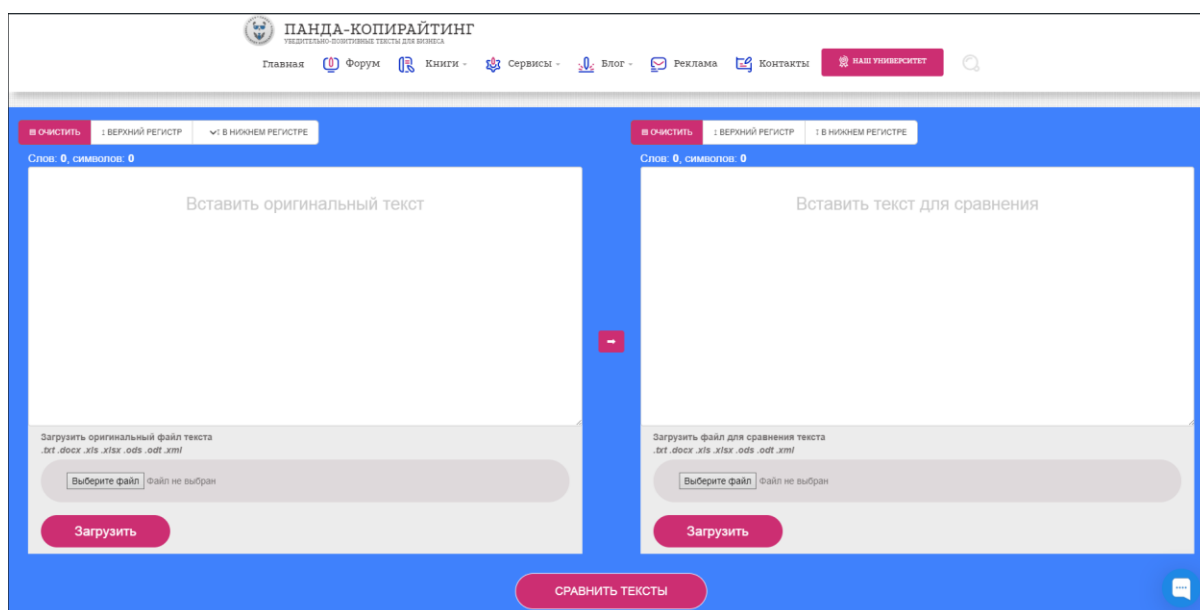


Рисунок 9 - Интерфейс страницы “Сравнение 2 текстов на схожесть”

Для тестирования сервиса взяли два различных текста. Первый текст описывал рельеф Алтайского края, второй - полезные ископаемые в Алтайском крае (Рисунок 10).

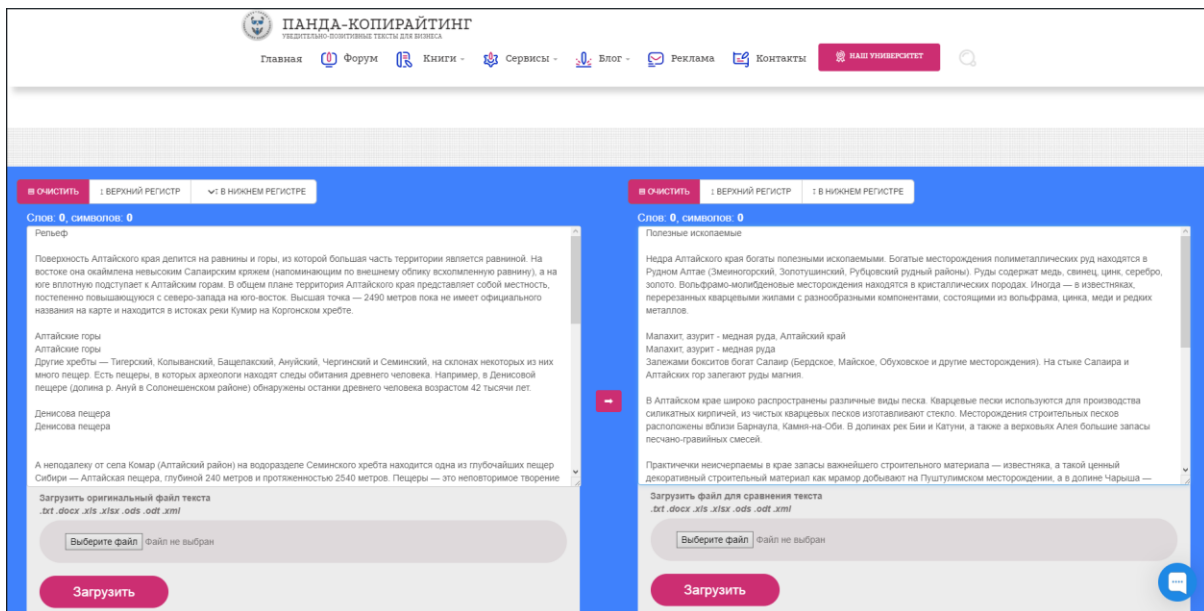


Рисунок 10 - Тестирование сервиса “Сравнение 2 текстов на схожесть”

На основе полученных результатов (Рисунок 11), можно сделать вывод, что тексты сравниваются посимвольно и никакого семантического анализа не производится. Т.е. если не вчитываться в текст, то невозможно понять, что и в одном, и в другом тексте речь идёт об Алтайском крае. Стоит отметить, что смысловой анализ не входит в задачи данного сервиса. Авторы работы предлагают лишь сравнение текстов.

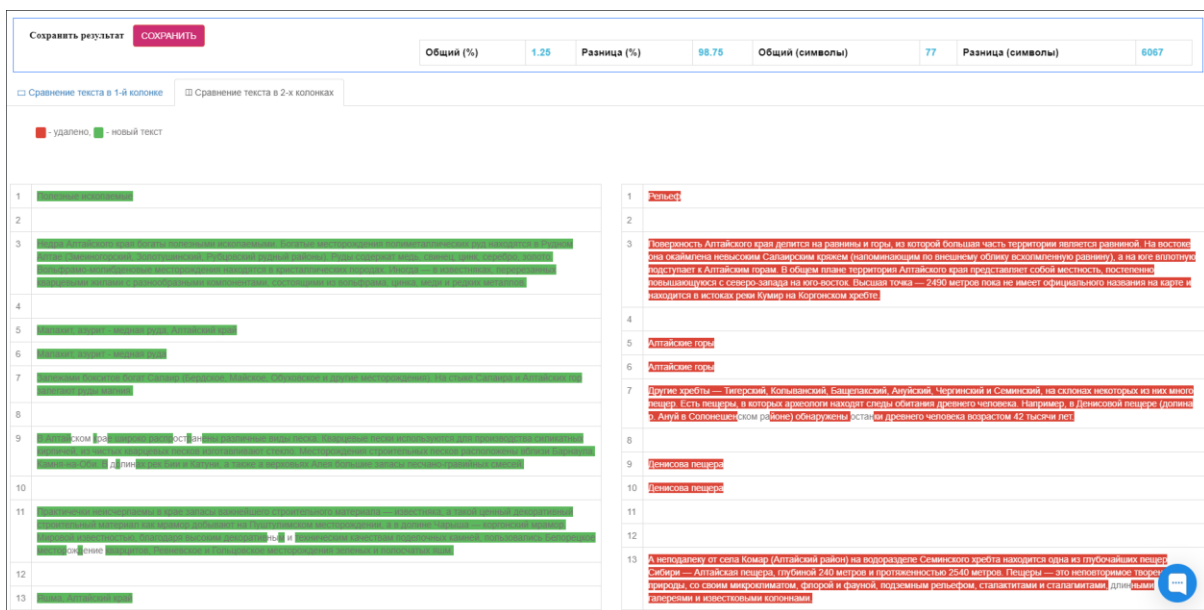


Рисунок 11 - Результат работы сервиса “Сравнение 2 текстов на схожесть”

### 1.3.4 Программный продукт “Miratext - биржа копирайтеров”

Ещё одним инструментом для работы с текстом является web-приложение “Miratext - биржа копирайтеров”. Функционал сайта представлен также большим набором средств для анализа текста. Но этот сервис предоставляет возможность работы одновременно только с одним материалом.

Для тестирования воспользуемся только двумя функциями сервиса - полный семантический анализ текста и контроль ключевых слов.

Алгоритм работы для полного семантического анализа текста весьма примитивен, достаточно вставить текст вручную или добавить ссылку на текст и воспользоваться кнопкой анализировать.

Воспользуемся вторым вариантом - добавим ссылку на источник (Рисунок 12).

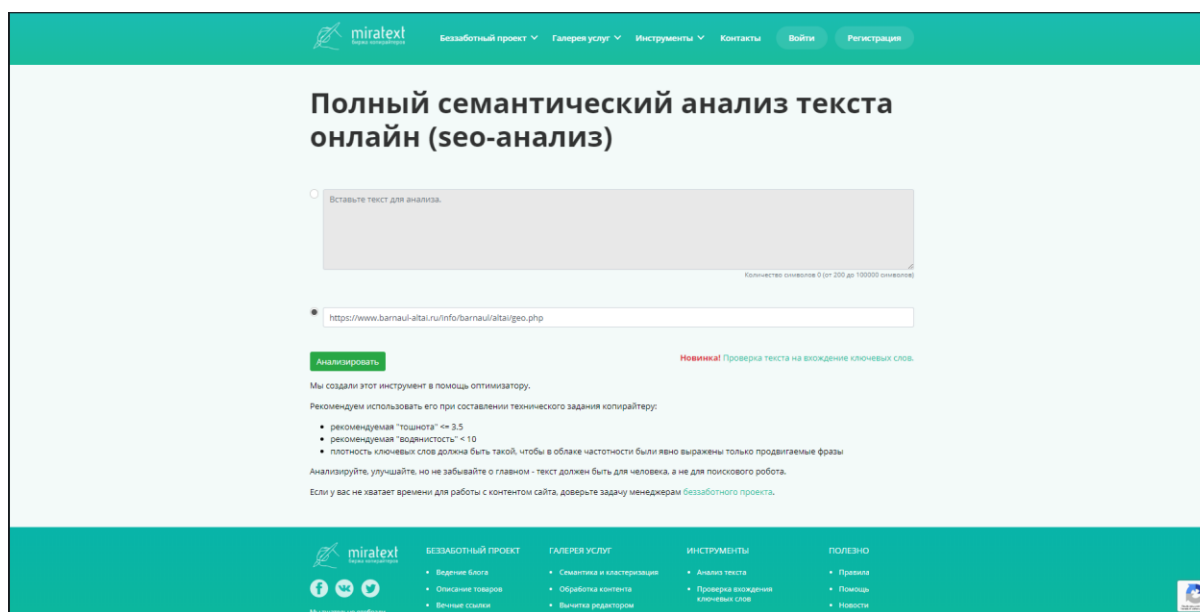
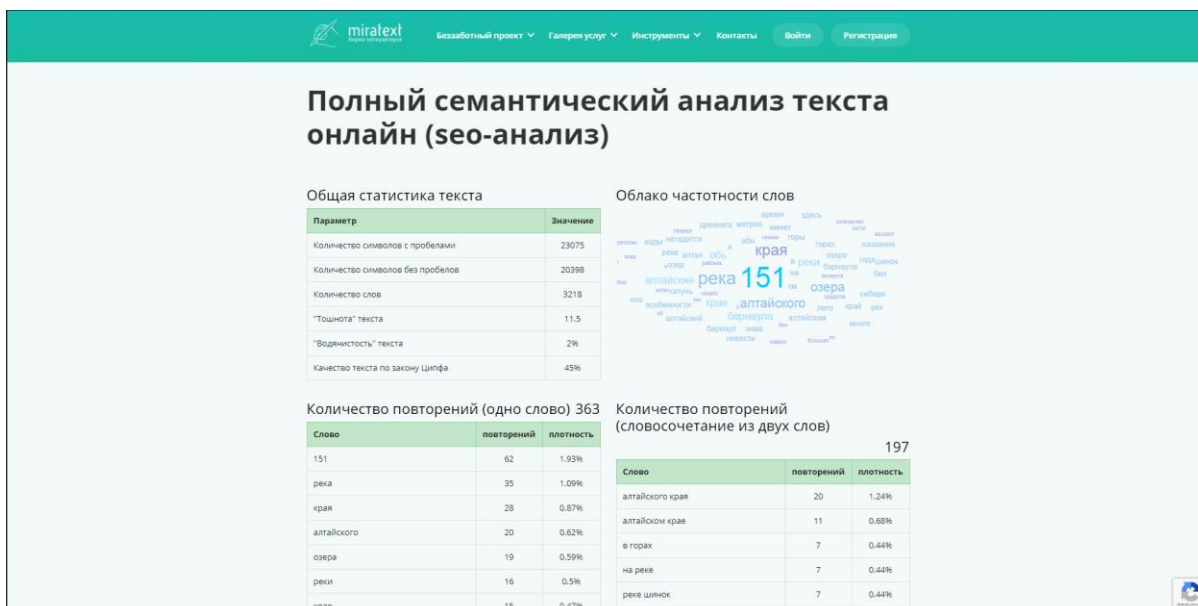


Рисунок 12 - Тестирование сервиса “Полный семантический анализ текста онлайн”

По результатам анализа (рисунок 13) пользователю предоставляется различные статические данные, такие как количество повторений (одно слово), количество повторений (словосочетание из двух слов), тошнота и др.



Кроме такого анализа можно выполнить проверку текста на вхождение ключевых слов. Для этого необходимо в текстовое поле добавить текст и далее ввести слова, которые нужно найти (совпадение может быть точным, разбавленным или типа LSI). Процесс поиска запускается с помощью соответствующей кнопки (Рисунок 14).

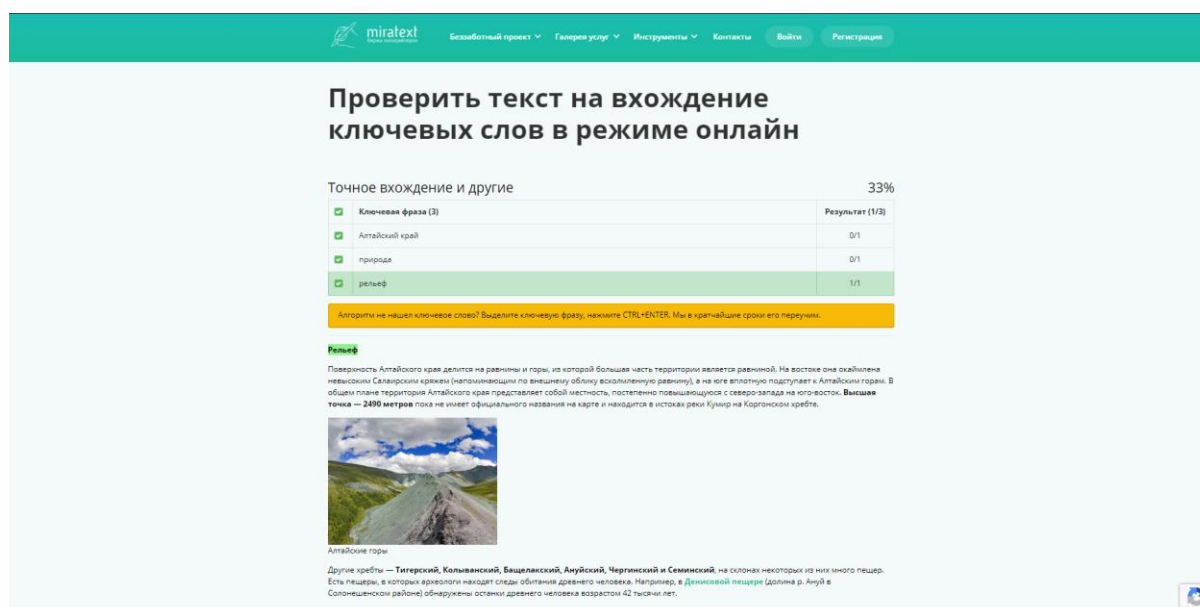


Рисунок 15 - Результат тестирования сервиса “Контроль ключевых слов”

Исходя из анализа различных ресурсов для работы с текстом, можем сделать вывод, что на данный момент в сети Интернет нет ПО, полностью отвечающего критериям поставленной задачи.

## 1.4 Функциональные возможности разрабатываемого приложения

На основе анализа предметной области, а также изучения различных инструментов для работы с текстами, в том числе учебными материалами, определим необходимую функциональность будущего веб-приложения.

1. Загрузка учебных материалов в формате docx, doc, txt или путём ручного набора текста:
  - 1.1. Ввод названия учебного материала (для авторизованного пользователя);
  - 1.2. Ввод текста учебного материала в textarea;
2. Работа со словами;
  - 2.1 Просмотр ключевых слов и слов-кандидатов в предметный тезаурус;
  - 2.2 Удаление ключевых слов;
  - 2.3 Расширение списка ключевых слов за счет добавления из списка слов-кандидатов.
3. Просмотр результатов анализа учебных материалов на совместимость;
4. Добавление одного документа (текста) и выделение для него ключевых слов с возможностью удаления;

5. Регистрация пользователей:
  - 5.1. Ввод логина;
  - 5.2. Ввод пароля;
  - 5.3. Ввод пароля повторно;
  - 5.4. Ввод почтового адреса;
6. Авторизация пользователей:
  - 6.1. Ввод логина;
  - 6.2. Ввод пароля;
7. Восстановление пароля пользователя:
  - 7.1. Ввод почтового адреса;
  - 7.2. Ввод нового пароля;
  - 7.3. Повторный ввод нового пароля;
8. Просмотр загруженных файлов для зарегистрированного пользователя:
  - 8.1. Удаление файла;
  - 8.2. Удаление ключевых слов в просматриваемом документе;
  - 8.3. Скачивание файла;
9. Просмотр истории анализов для зарегистрированного пользователя;
10. Выход из аккаунта;
11. Изменение личных данных (для зарегистрированного пользователя):
  - 11.1. Изменение пароля.
12. Администрирование сайта.

## 2 Проектирование web-приложения

### 2.1 Структура приложения

Исходя из описанной выше функциональности определим структуру будущего сайта, которая обеспечит удобную и интуитивно понятную работу с сервисом.

Сайт должен содержать следующие страницы:

- 1) Главная страница;
- 2) Страница регистрации;
- 3) Страница авторизации;
- 4) Страница восстановления пароля;
- 5) Личный кабинет;
- 6) Страница добавления для анализа одного материала;
- 7) Страница просмотра/удаления ключевых слов одного материала и результата анализа;
- 8) Страница добавления для анализа двух материалов;
- 9) Страница просмотра/удаления ключевых слов для двух материалов и результата анализа;
- 10) Страница просмотра/удаления загруженных материалов;
- 11) Страница детального просмотра одного загруженного материала;
- 12) Страница просмотра истории проведенных анализов;
- 13) Страница администратора сайта.

Общая структура сайта представлена на рисунке 16.





- добавление одного/двух материала(-ов) для анализа;
- просмотр/ удаление слов для одного/двух материалов;
- просмотр результатов анализа на совместимость.

Для зарегистрированного пользователя доступны возможности незарегистрированного пользователя, а также просмотр/удаление загруженным материалов и просмотр истории проведенных анализов.

Кроме этого для контроля работы приложения, а также возможности вносить изменения в систему не программно, введем еще один тип пользователя - администратора сайта. Администратор сайта нужен не только для вышеописанных задач, но и для взаимодействия с пользователями. Так пользователи смогут сообщить об ошибках системы или внести какие-то предложения по улучшению сервиса.

### **2.2.2 Описание процесса регистрации и авторизации**

Для использования всех возможностей сайта пользователь должен пройти процесс регистрации. Наш сервис не предполагает хранение персональной информации, а также проведение коммерческих операций. Поэтому для регистрации пользователя было решено использовать минимум параметров, а именно логин, адрес электронной почты и пароль. Адрес электронной почты необходим для восстановления пароля в случае его утери пользователем. После регистрации пользователь попадает на главную страницу зарегистрированного пользователя.

Для входа в систему пользователь, прошедший регистрацию не в рамках текущей сессии, должен пройти процесс авторизации. Для этого необходимо ввести логин и пароль учетной записи.

### **2.2.3 Описание процесса работы с одним учебным материалом**

У пользователей системы может возникнуть потребность работы только с одним учебным материалом. Причем у зарегистрированного и незарегистрированного пользователи эти потребности могут различаться.

Неавторизованный пользователь может загрузить через соответствующую форму документ или ввести его вручную в поле `textarea` и получить список слов-кандидатов для данного документа. Слова-кандидаты - слова, которые могли попасть в предметный тезаурус, но были отсеяны алгоритмом.

Такая операция для зарегистрированного пользователя носит расширенный характер. Документ и выделенные слова предметного тезауруса сохраняются в библиотеке пользователя, а также пользователь может удалять определенные слова, если таковыми их не считает, или расширять их за счёт слов-кандидатов. Это же ему доступно и в случае просмотра документа из библиотеки. Но пользователю нужно будет заполнить обязательное поле “Наименование материала” для дальнейшего корректного отображения в библиотеке.

#### **2.2.4 Описание процесса работы с двумя учебными материалами**

Процесс работы пользователей с двумя материалами аналогичен работе с одним материалом, но для двух документов можно рассчитать внутреннюю и внешнюю совместимость текста. Для расчёта внешней совместимости предполагается загрузка (ручной набор) материалов различных дисциплин. Для расчёта внутренней совместимости - загрузка (ручной набор) отдельных тем одного ресурса.

Так как для разных видов анализа должна быть получена соответствующая им оценка, пользователю необходимо установить значение чекбокса в зависимости от проводимого анализа. Например, при расчете внутренней совместимости вместе с результатом анализа пользователь получит сообщение о том, какой диапазон значений считается оптимальным для внутренней совместимости текстов и попадает ли полученный результат анализа в этот диапазон. Аналогично и для внешней совместимости.

#### **2.2.4 Описание процесса работы с личным кабинетом**

В личном кабинете предполагается отображение библиотеки пользователя - комплекс загруженных пользователем материалов и истории проведенных анализов.

Работа с библиотекой перетекает в работу с отдельными файлами. Имеется

возможность удаления файла. Работа с историей - просмотр списка, включающего как анализ одного документа на ключевые слова, так и анализ двух документов на совместимость с указанием типа совместимости.

Кроме этого с помощью личного кабинета пользователь может изменить данные учетной записи, например, поменять пароль.

## 3 Выбор инструментов разработки

### 3.1 Клиентская часть

Для разработки пользовательского интерфейса использовались следующие средства.

Bootstrap (также известен как Twitter Bootstrap) — свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

### 3.2 Серверная часть

Для разработки серверной части веб-приложения использовались такие инструменты.

Python – высокоуровневый язык программирования общего назначения. Синтаксис языка минималистичен, легко читается и легок в изучении. Он является наиболее распространенным инструментом для работы с текстами. Также он имеет множество полезных библиотек.

Django — фреймворк для перфекционистов с дедлайнами. Свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC. Проект поддерживается организацией Django Software Foundation.

Jinja — это шаблонизатор для языка программирования Python. Он подобен шаблонизатору Django, но предоставляет Python-подобные выражения, обеспечивая исполнение шаблонов в песочнице. Это текстовый шаблонизатор, поэтому он может быть использован для создания любого вида разметки, а также исходного кода. Лицензирован под BSD лицензией. Шаблонизатор Jinja позволяет настраивать теги, фильтры, тесты и глобальные переменные. Также, в отличие от шаблонизатора Django, Jinja позволяет конструктору шаблонов вызывать функции с аргументами на объектах.

AJAX, Ajax (от англ. Asynchronous Javascript and XML — «асинхронный JavaScript и XML») — подход к построению интерактивных пользовательских

интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером. В результате, при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее.

MySQL — свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией.

### **3.3 Дополнительные инструменты**

Кроме средств для разработки клиентской и серверной части, для более удобной работы с проектом использовались следующие инструменты.

GitHub — крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Веб-сервис основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub, Inc (ранее Logical Awesome). Сервис бесплатен для проектов с открытым исходным кодом и (с 2019 года) небольших частных проектов, предоставляя им все возможности (включая SSL).

Atom — бесплатный текстовый редактор с открытым исходным кодом для Linux, macOS, Windows с поддержкой плагинов, написанных на Node.js, и встраиваемых под управлением Git. Большинство плагинов имеют статус свободного программного обеспечения, разрабатываются и поддерживаются сообществом.

Apache HTTP-сервер — свободный веб-сервер. Apache является кроссплатформенным ПО, поддерживает операционные системы Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS.

Основными достоинствами Apache считаются надёжность и гибкость конфигурации. Он позволяет подключать внешние модули для предоставления данных, использовать СУБД для аутентификации пользователей, модифицировать сообщения об ошибках и т. д. Поддерживает IPv6.

MySQL Workbench - унифицированный визуальный инструмент для архитекторов, разработчиков и администраторов баз данных. MySQL Workbench

предоставляет моделирование данных, разработку SQL и комплексные инструменты администрирования для настройки сервера, администрирования пользователей, резервного копирования и многого другого. MySQL Workbench доступен в Windows, Linux и Mac OS X.

### **3.4 Библиотека для расчёта совместимости учебных материалов**

Основной задачей работы является расчёт внутренней и внешней совместимости учебных материалов. Для реализации основной функциональности приложения было решено использовать модуль для работы с естественными языками, разработанный Нефёдовой Анастасией в рамках выпускной квалификационной работы.

Модуль имеет следующие функциональные возможности:

- Выделение слов из текста;
- Вычисление TF-IDF для каждого слова;
- Составление тезауруса, используя закон Ципфа и закон Хикса;
- Проведение анализа совместимости учебных материалов.

Данное программное обеспечение позволяет в полной мере реализовать требования к разрабатываемой нами системе. Также стоит отметить, что модуль написан на языке программирования Python, что позволяет легко интегрировать его в нашу задачу.

## 4 Архитектура приложения

### 4.1 База данных

В ходе анализа предметной области и требований к функционалу были выделены следующие сущности базы данных:

- пользователь;
- документ;
- результат анализа;
- ключевое слово.

Разработана реляционная база данных. Концептуальная модель базы представлена на рисунке 17.

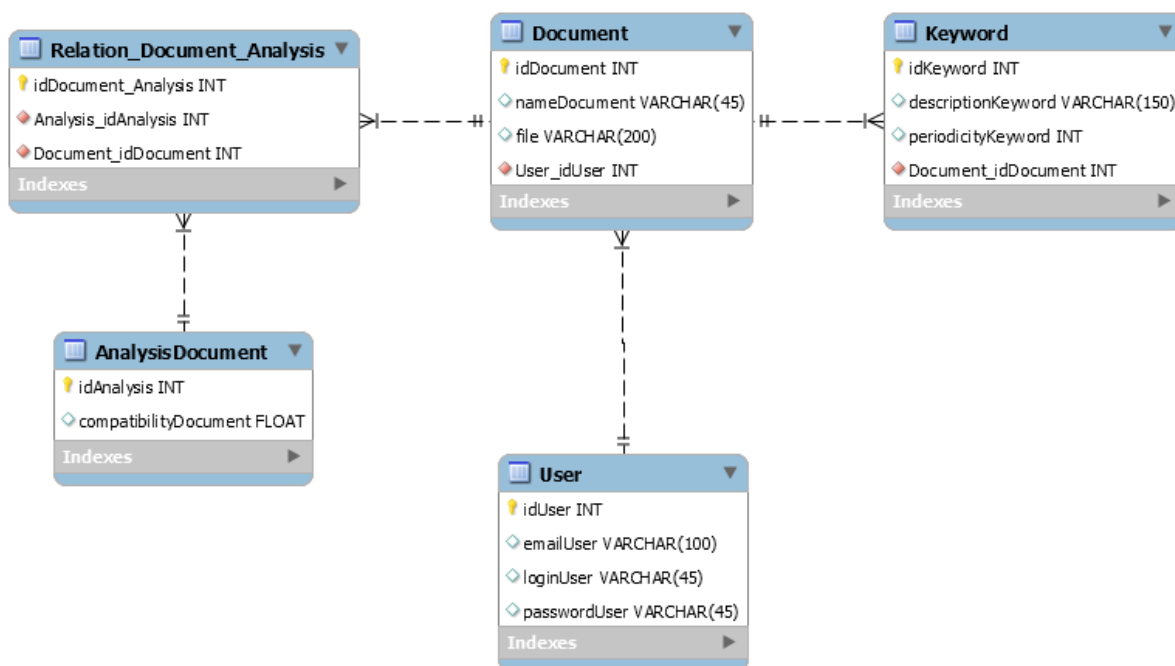


Рисунок 17 - Концептуальная модель базы данных

Ниже представлено подробное описание сущностей.

Сущность User (пользователь) содержит информацию о зарегистрированном пользователе системы.

Таблица 1 - Описание сущности User (пользователь)

Поле	Тип данных	Назначение
idUser	int	первичный ключ
emailUser	varchar(100)	адрес электронной

		почты
loginUser	varchar(45)	логин
passwordUser	varchar(45)	пароль

Сущность Document (документ) содержит информацию о загруженном материале для анализа на совместимость.

Таблица 2 - Описание сущности Document (документ)

Поле	Тип данных	Назначение
idDocument	int	первичный ключ
nameDocument	varchar(45)	наименование
file	varchar(200)	путь к файлу на сервере
User_idUser	int	внешний ключ таблицы "User"

Сущность Keyword (ключевое слово) содержит информацию о ключевых словах материала.

Таблица 3 - Описание сущности Keyword (ключевое слово)

Поле	Тип данных	Назначение
idKeyword	int	первичный ключ
descriptionKeyword	varchar(150)	описание ключевого слова
periodicityKeyword	int	частота слова в материале
Document_idDocument	int	внешний ключ таблицы "Document"

Сущность AnalysisDocument (анализ материалов) содержит результат анализа двух материалов.

Таблица 4 - Описание сущности AnalysisDocument (анализ материалов)

Поле	Тип данных	Назначение
------	------------	------------



idAnalysisDocument	int	первичный ключ
compatibilityDocument	float	совместимость в процентах

Сущность Relation\_Document\_Analysis (отношение документ - анализ) - расшивочная таблица для отношения документ-анализ.

Таблица 5 - Описание сущности Relation\_Document\_Analysis (отношение документ - анализ)

Поле	Тип данных	Назначение
idDocument_Analysis	int	первичный ключ
Document_idDocument	int	внешний ключ таблицы "Document"
Analysis_idAnalysis	int	внешний ключ таблицы "AnalysisDocument "

## 4.2 Описание архитектуры веб-приложения

Приложение имеет трехуровневую архитектуру (рисунок 18). В функцию клиентской части входит только интерактивное взаимодействие с пользователем, бизнес-логика вынесена на сервер приложения, который, собственно, и обеспечивает формирование запросов к базе данных, передаваемых на выполнение серверу базы данных.

К достоинствам такой архитектуры стоит отнести: масштабируемость, высокую безопасность и надежность. К недостаткам - сложность администрирования и обслуживания, высокие требования к производительности серверов приложений и серверов базы данных, высокие требования к скорости канала между сервером базы данных и сервером приложений.

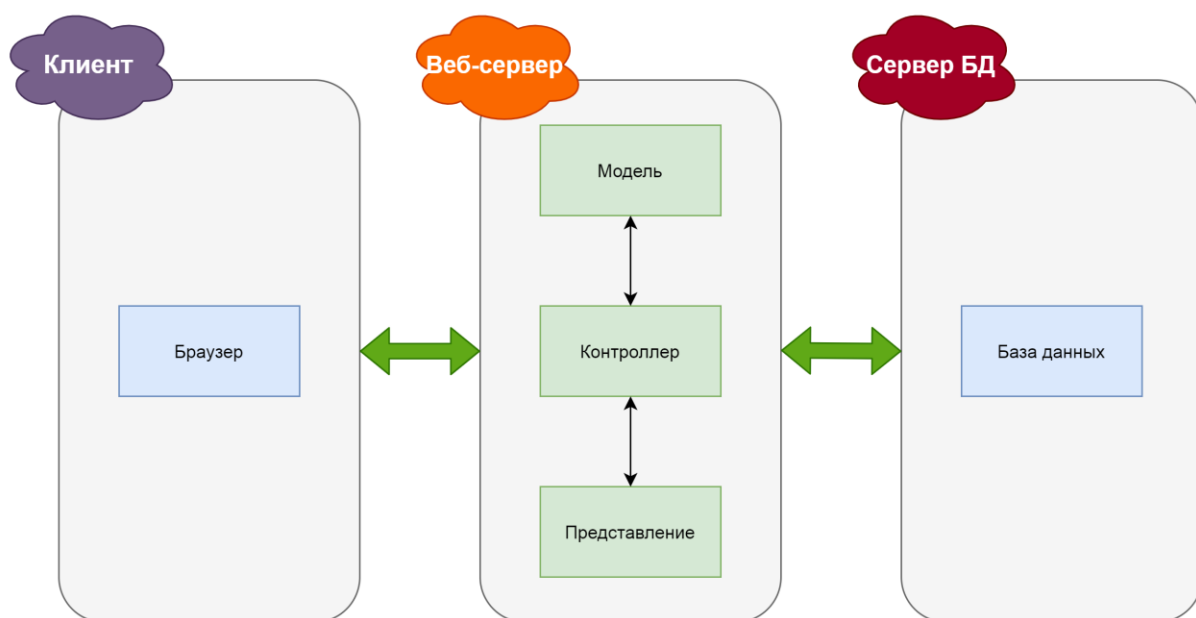


Рисунок 18 - Архитектура приложения

Уровень приложения, реализующий бизнес-логику системы спроектирован на основе паттерна проектирования MVC (Model-View-Controller) (рисунок 19). Model-View-Controller («Модель-Представление-Контроллер») — схема разделения данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: модель, представление и контроллер — таким образом, что модификация каждого компонента может осуществляться независимо.

Модель (Model) предоставляет данные и реагирует на команды контроллера, изменяя своё состояние.

Представление (View) отвечает за отображение данных модели пользователю, реагируя на изменения модели.

Контроллер (Controller) интерпретирует действия пользователя, оповещая модель о необходимости изменений.

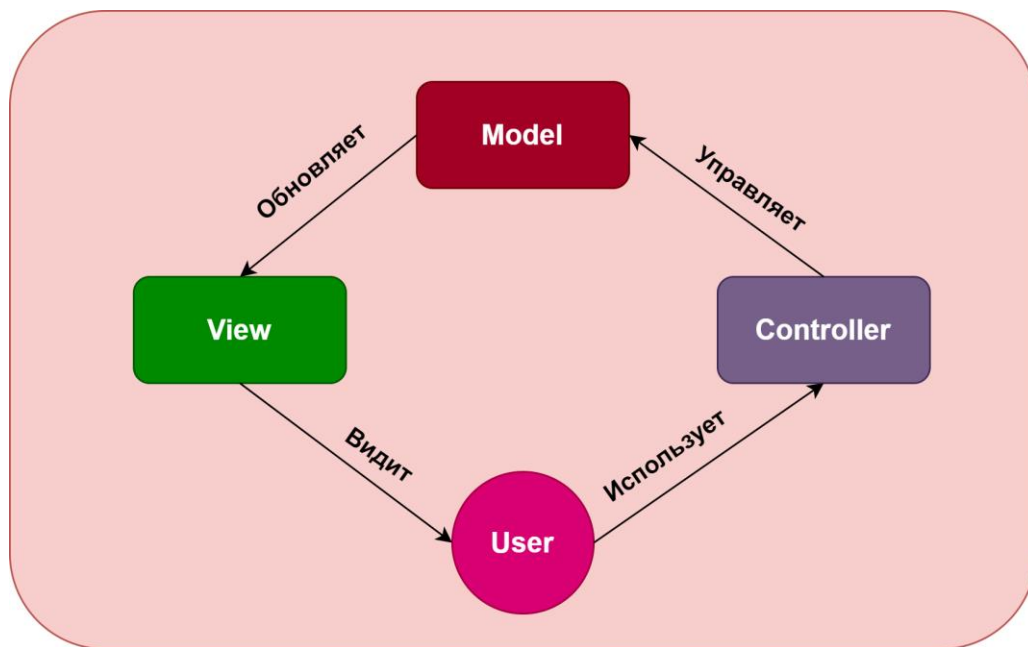


Рисунок 19 - Шаблон проектирования MVC

### 4.3 Интерфейс приложения

Интерфейс страниц приложения представлен на рисунках 20-32.

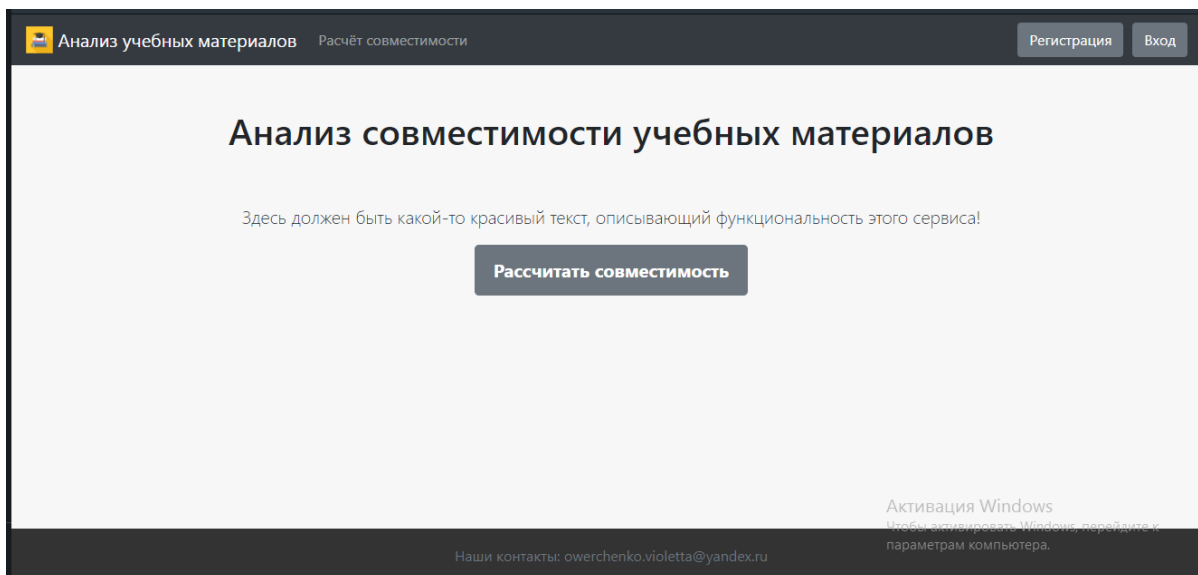


Рисунок 20 - Главная страница

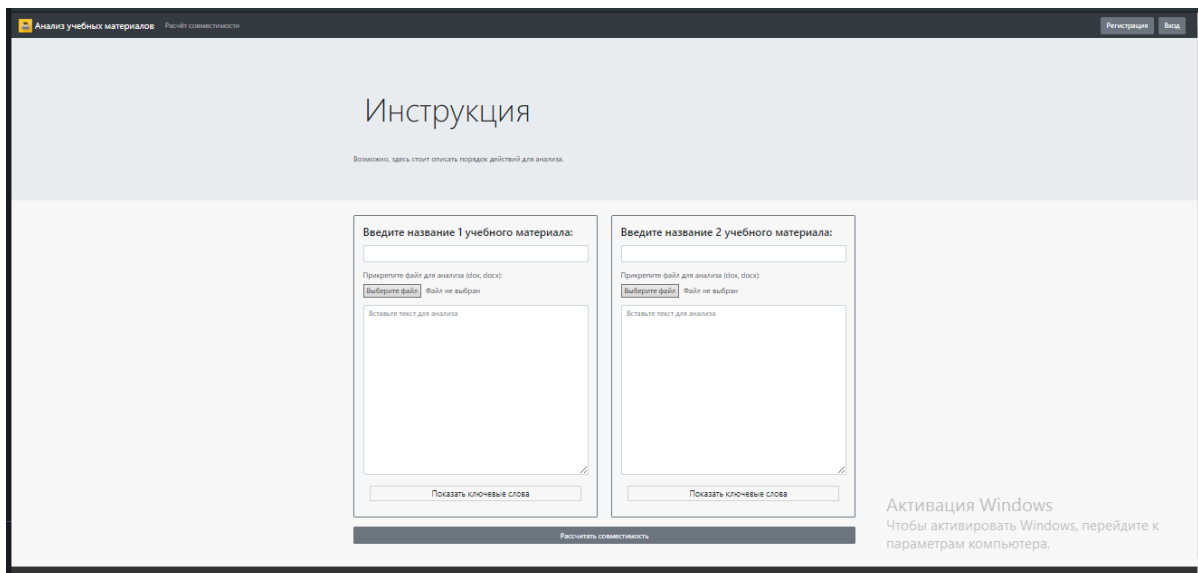


Рисунок 21 - Страница для расчёта совместимости 2 материалов

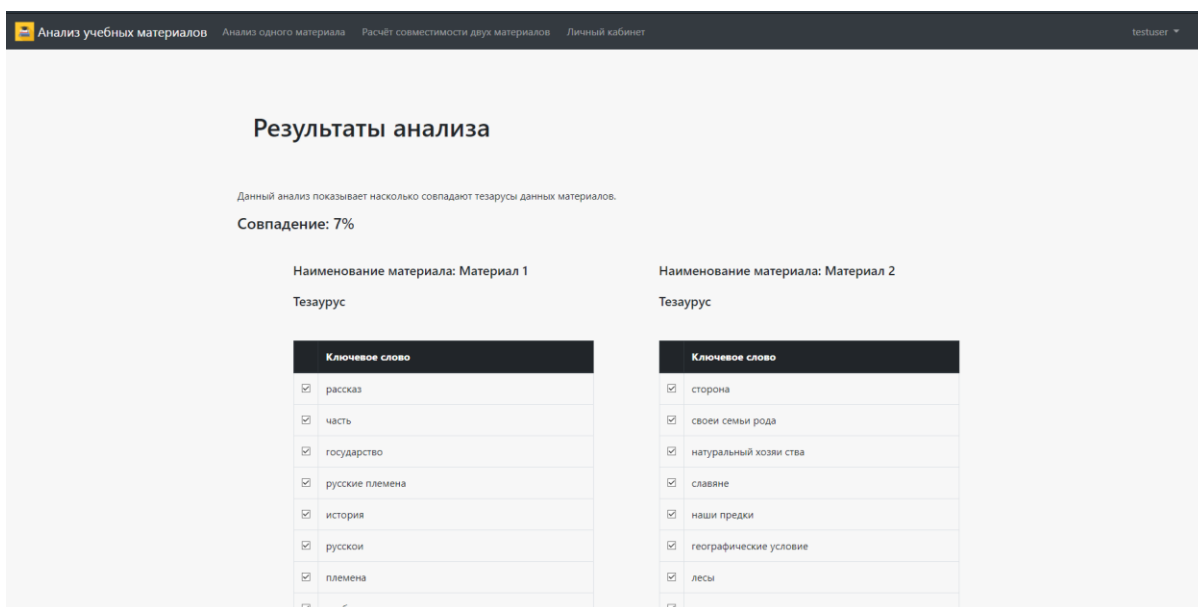


Рисунок 22 - Страница для просмотра результата анализа 2 материалов


Анализ учебных материалов

Расчёт совместимости

Регистрация

Вход

Авторизация



Логин

Пароль

Войти

[Забыли пароль?](#)

[Создать аккаунт](#)

Активация Windows  
Чтобы активировать Windows, перейдите к параметрам компьютера.

Наши контакты: owerchenko.violetta@yandex.ru

Рисунок 23 - Страница авторизации пользователя

Анализ учебных материалов

Расчёт совместимости

Регистрация

Вход

Восстановление пароля

Для восстановления пароля введите email:

Email

Сброс пароля

Активация Windows  
Чтобы активировать Windows, перейдите к параметрам компьютера.

Наши контакты: owerchenko.violetta@yandex.ru

Рисунок 24 - Страница восстановления пароля

Анализ учебных материалов

Расчёт совместимости

Регистрация

Вход

Восстановление пароля

Мы отправили вам по электронной почте инструкции по установке пароля. Если они не прибыли в течение нескольких минут, проверьте папку со спамом.

Активация Windows  
Чтобы активировать Windows, перейдите к параметрам компьютера.

Наши контакты: owerchenko.violetta@yandex.ru

Рисунок 25 - Информационная страница для восстановления пароля

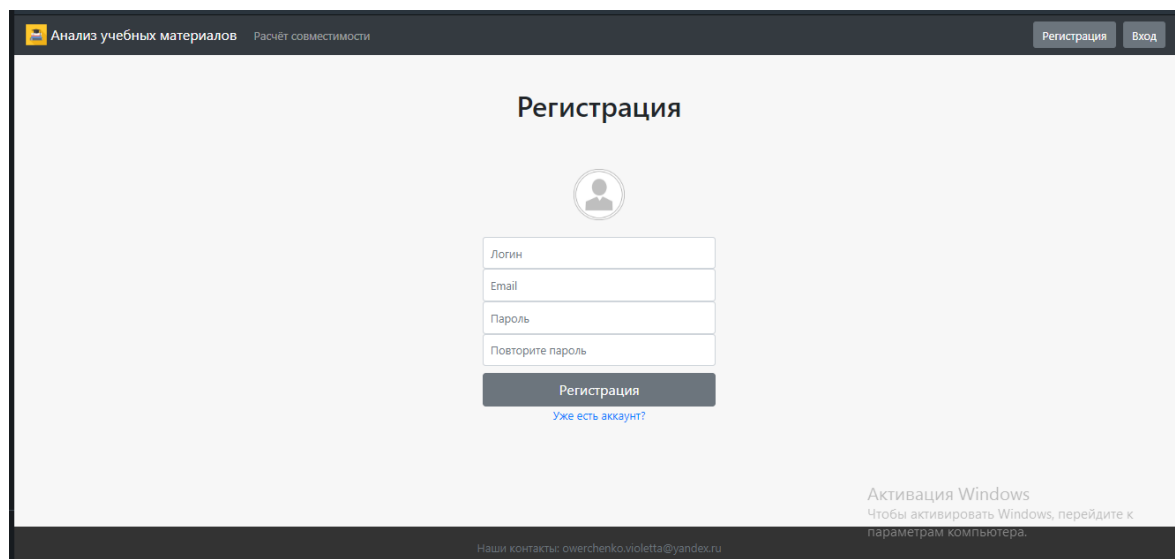


Рисунок 26 - Страница регистрации

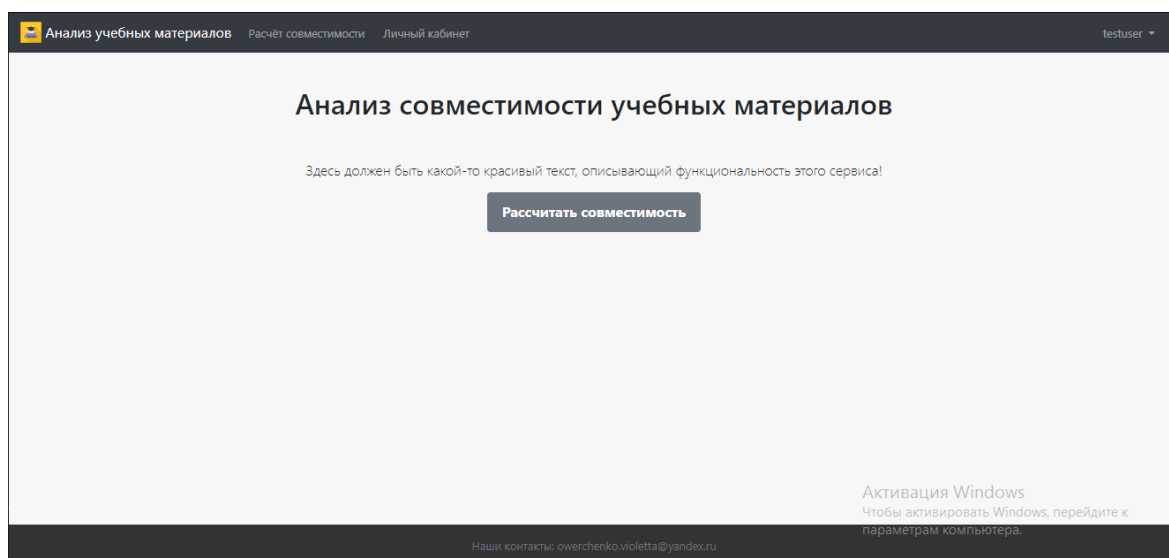


Рисунок 27 - Главная страница авторизованного пользователя

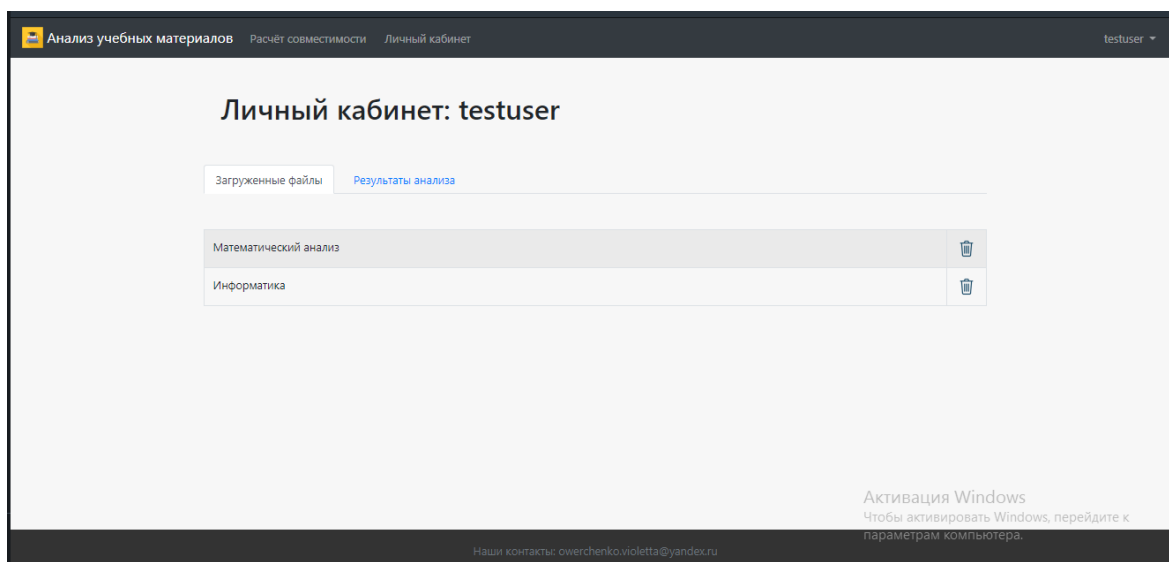


Рисунок 28 - Страница “Личный кабинет”. Загруженные материалы

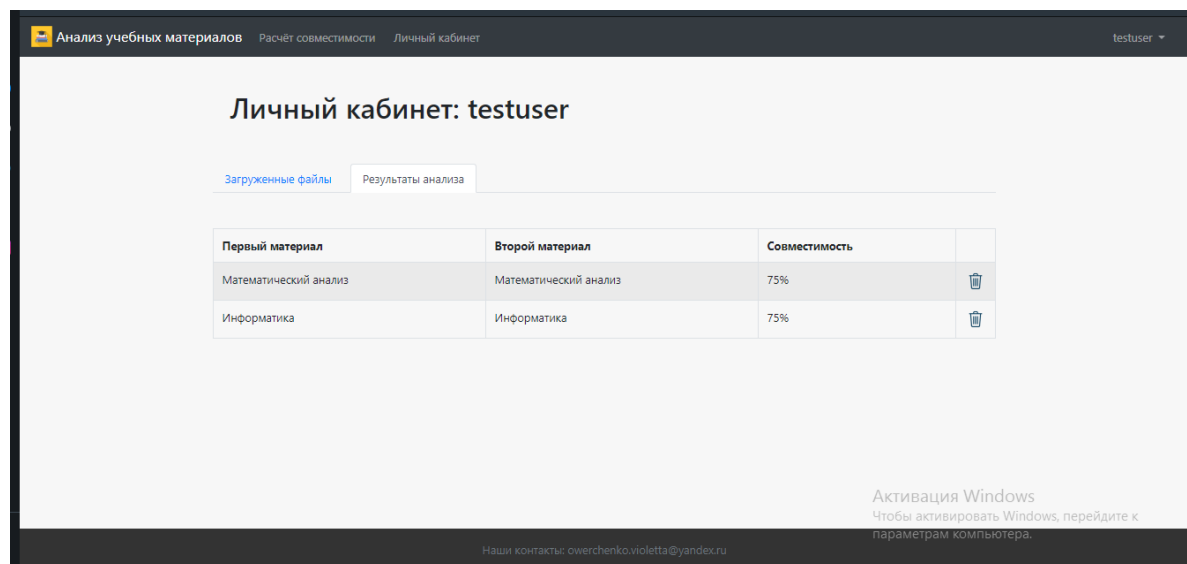


Рисунок 29 - Страница “Личный кабинет”. Результаты анализа

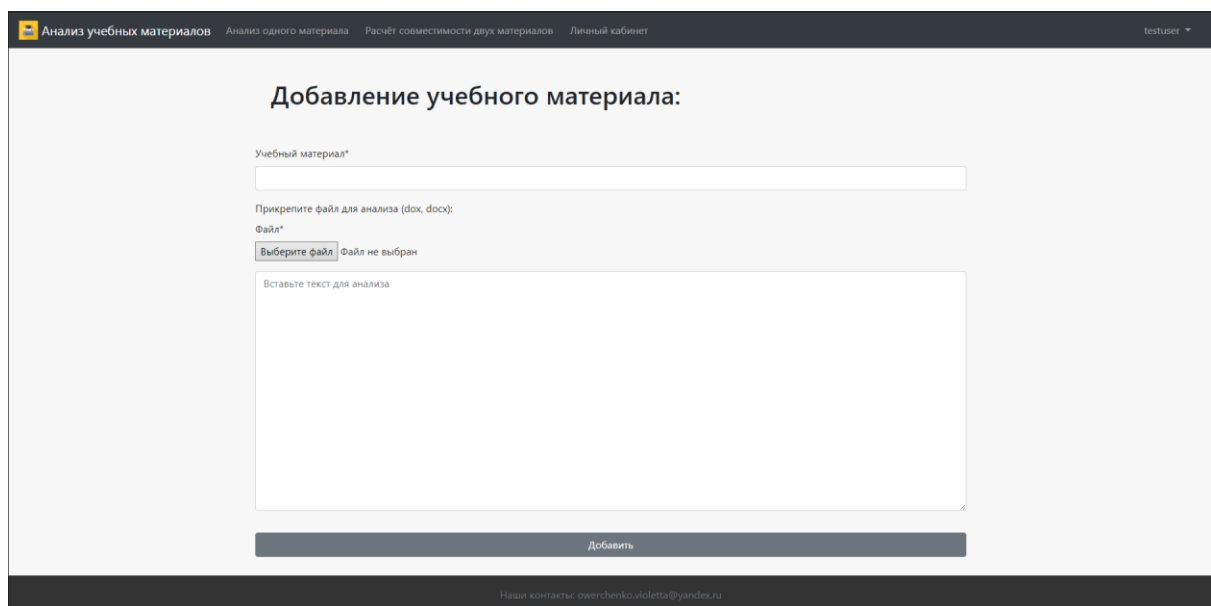


Рисунок 30 - Страница “Добавление одного материала”

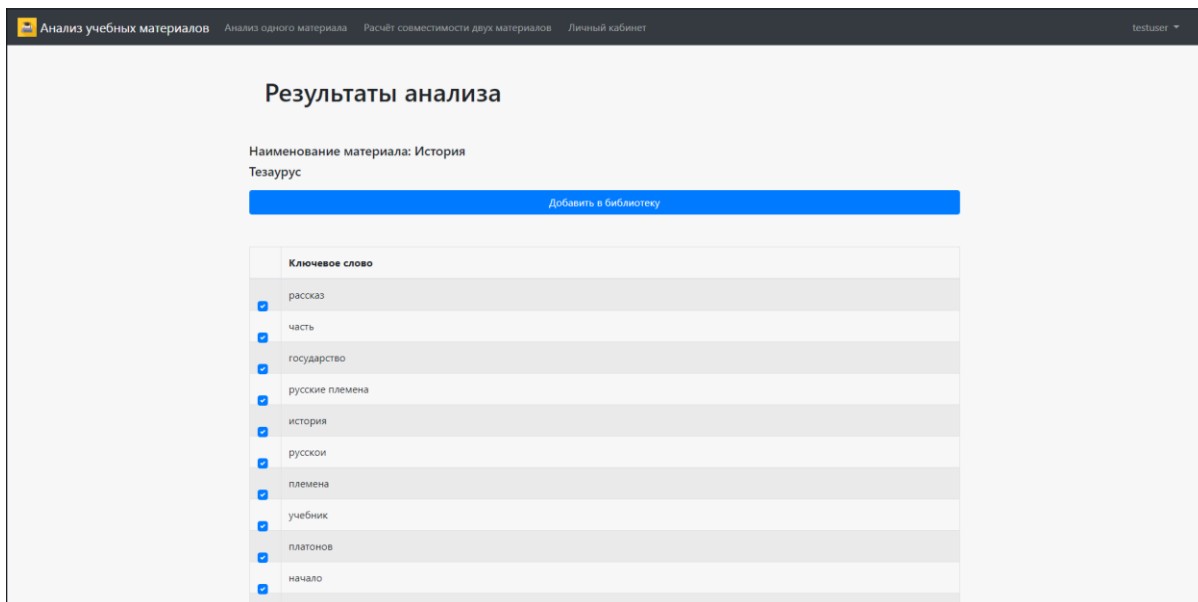


Рисунок 31 - Страница “Результат анализа одного материала”

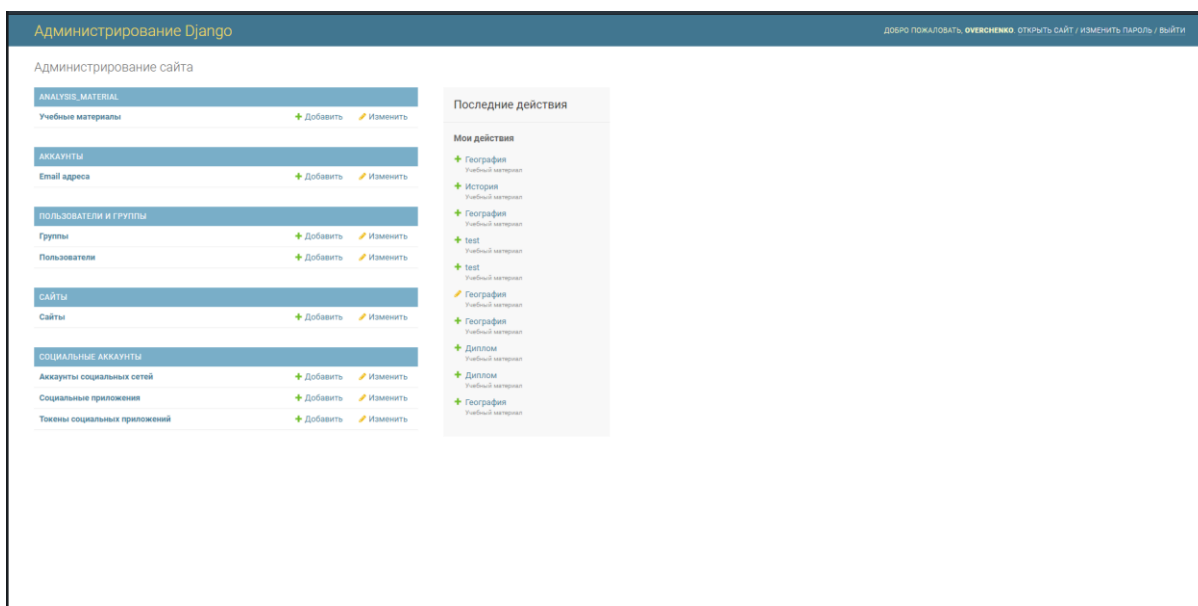


Рисунок 32 - Страница “Администрирование”



## 5 Тестирование

Как известно, любое программное обеспечение во время разработки и непосредственно перед выпуском требует комплексного тестирования. Это позволяет предоставить пользователю качественный и надежный продукт. Для максимально эффективной разработки программы тестирование лучше проводить параллельно с процессом написания кода. Это даёт возможность своевременно исправлять все обнаружившиеся баги, и тем самым не только повышать качество программного обеспечения, но и сокращать трудозатраты программиста.

На сегодняшний день существует множество средств для проведения комплексных тестов. Эти средства различаются в зависимости от вида тестирования.

Так выделяют следующие виды тестирования:

- по объекту тестирования;
- по знанию внутреннего строения системы;
- по степени автоматизации;
- по степени изолированности;
- по времени проведения тестирования;
- по признаку позитивности сценариев;
- по степени подготовленности к тестированию.

В свою очередь каждый вид содержит в себе несколько типов.

Такая широкая классификация объясняется тем, что этап тестирования при разработке программного обеспечения является одним из ключевых и значимых.

Для автоматизированного тестирования нашего приложения мы использовали unit-тестирование (модульное тестирование). Этот тип тестирования предназначен для тестирования отдельных модулей приложения.

Так как для написания приложения мы использовали фреймворк Django, то и для его тестирования воспользуемся средствами, предоставляемыми этим фреймворком. Для тестирования приложений Django использует стандартную библиотеку Python unittest. Данная библиотека позволяет проводить не только юнит-тестирование, но и интеграционное тестирование. Это дает возможность

имитировать URL-запросы, добавлять тестовые данные и проводить проверку выходных данных приложения. Также Django предоставляет API и средства для подключения других фреймворков тестирования, например, фреймворка Selenium для имитации действий пользователя в браузере.

Для создания теста необходимо наследоваться от одного из классов тестирования Django, а потом реализовать конкретные методы проверки. Стоит отметить, что тесты должны быть написаны только для кода, не касающегося стандартных библиотек и функциональности, предоставляемых Python и Django.

Для тестирования нашего приложения мы ввели три уровня тестирования, а именно:

- тестирование моделей;
- тестирование форм;
- тестирование представлений.

Рассмотрим каждый уровень подробнее.

В качестве примера тестирования моделей возьмем модель Document (Рисунок 33).

```
class Document(models.Model):
    user = models.ForeignKey(User, verbose_name="Пользователь", on_delete=models.SET_NULL, null=True, blank=True)
    name_document = models.CharField("Учебный материал", max_length=200)
    path_file = models.FileField("Файл", upload_to='documents/')
    uploaded_at = models.DateTimeField("Время загрузки", auto_now_add=True)

    def __str__(self):
        return self.name_document

    def get_absolute_url(self):
        return reverse("document_detail", kwargs={"pk": self.pk})

    class Meta:
        verbose_name = "Учебный материал"
        verbose_name_plural = "Учебные материалы"
```

Рисунок 33 - Модель "Document"

Нам необходимо протестировать текстовые метки всех полей, так мы будем уверены в том, что эти метки содержат нужные значения. Django создаст поле заданной длины, а значит тесты проверят необходимый размер поля, а также содержимое этого поля (Рисунок 34).

```

class DocumentModelTest(TestCase):

    @classmethod
    def setUpTestData(cls):
        Document.objects.create(name_document='История', path_file='documents/1_2_главы.docx')

    def test_name_document_label(self):
        document = Document.objects.get(id=1)
        field_label = document._meta.get_field('name_document').verbose_name
        self.assertEqual(field_label, 'Учебный материал')

    def test_uploaded_at_label(self):
        document = Document.objects.get(id=1)
        field_label = document._meta.get_field('uploaded_at').verbose_name
        self.assertEqual(field_label, 'Время загрузки')

    def test_path_file_label(self):
        document = Document.objects.get(id=1)
        field_label = document._meta.get_field('path_file').verbose_name
        self.assertEqual(field_label, 'Файл')

    def test_name_document_max_length(self):
        document = Document.objects.get(id=1)
        max_length = document._meta.get_field('name_document').max_length
        self.assertEqual(max_length, 200)

```

Рисунок 34 - Методы для тестирования полей модели “Document”

Также необходимо протестировать собственные метода. Для этого проверяем, что имена объектов имеют значения “Name Document”, а URL-адрес, по которому получаем экземпляр модели Document, такой как ожидается (Рисунок 35).

```

def test_object_name_is_name_document(self):    # проверка для метода __str__
    document = Document.objects.get(id=1)
    expected_object_name = document.name_document
    self.assertEqual(expected_object_name, str(document))

def test_get_absolute_url(self):
    document = Document.objects.get(id=1)
    self.assertEqual(document.get_absolute_url(), '/1/')

```

Рисунок 35 - Методы для тестирования собственных методов модели “Document”

Тестирование форм имеет тот же смысл, что и тестирование моделей - нужно проверить всё, что не относится к фреймворку и подключаемым библиотекам. То есть необходимо написать тесты, проверяющие наличие определенных полей с соответствующими им метками и текстами. Также протестировать все методы, осуществляющие валидацию форм и написанные именно нами.

В качестве примера рассмотрим форму создания документа (Рисунок 36).

```
class DocumentForm(forms.ModelForm):
    class Meta:
        model = Document
        fields = ['name_document', 'path_file']
```

Рисунок 36 - Форма для создания документа

Для данной формы необходимо проверить текстовые метки для полей “name\_document”, “path\_file” (Рисунок 37).

```
class DocumentFormTest(TestCase):

    def test_document_form_name_field_label(self):
        form = DocumentForm()
        self.assertTrue(form.fields['name_document'].label == None or form.fields['name_document'].label == 'Учебный материал')

    def test_document_form_path_field_label(self):
        form = DocumentForm()
        self.assertTrue(form.fields['path_file'].label == None or form.fields['path_file'].label == 'Файл')
```

Рисунок 37 - Методы для проверки текстовых меток формы

Поскольку данная форма не содержит собственных методов валидации полей, то в дополнительных тестах нет необходимости.

Ещё одним уровнем проверки является - тестирование представлений. Для тестирования отображений используется так называемый тестовый клиент - Django Client. Этот класс позволяет имитировать GET и POST запросы и проверять полученные ответы. Про ответы можно узнать практически всё, начиная от итоговых заголовков и кодов статуса до применяемых шаблонов, используемых в рендеринге, и даже контекста, передаваемого в шаблон. Также возможно проверить URL-адреса и коды статусов.

Для примера протестируем представление, которое возвращает список всех учебных материалов, загруженных зарегистрированным пользователем (Рисунок 38). Его можно увидеть по адресу /personal\_account/.

```
class PersonalAccountViews(ListView):
    """ Личный кабинет: список документов, список анализов """
    model = Document
    template_name = "analysis_material/personal_account.html"
    paginate_by = 10

    def get_queryset(self):
        return Document.objects.filter(user=self.request.user)
```

Рисунок 38 - Класс, описывающий представление “Личный кабинет”

Так как данное представление использует классическое отображение ListView, то единственное, что необходимо протестировать - переход к нашему представлению по указанному URL-адресу. Итак, нужно проверить, что пользователь авторизованный и выводится список материалов, загруженных этим пользователем (Рисунок 39).

```
class DocumentByUserListViewTest(TestCase):

    def setUp(self):
        # Создание двух пользователей
        test_user1 = User.objects.create_user(username='testuser1', password='12345')
        test_user1.save()
        test_user2 = User.objects.create_user(username='testuser2', password='12345')
        test_user2.save()

        number_of_documents = 13
        for document_num in range(number_of_documents):
            if document_num % 2:
                usertest = test_user1
            else:
                usertest = test_user2
            Document.objects.create(user = usertest, name_document = 'Материал %s' % document_num, path_file='Файл %s' % document_num,)

    def test_logged_in_uses_correct_template(self):
        login = self.client.login(username='testuser1', password='12345')
        resp = self.client.get(reverse('personal_account'))

        # Проверка что пользователь залогинился
        self.assertEqual(str(resp.context['user']), 'testuser1')
        # Проверка ответа на запрос
        self.assertEqual(resp.status_code, 200)

        # Проверка того, что мы используем правильный шаблон
        self.assertTemplateUsed(resp, 'analysis_material/personal_account.html')

        #Подтверждение, что все документы принадлежат testuser1
        for document in resp.context['document_list']:
            self.assertEqual(resp.context['user'], document.user)
```

Рисунок 39 - Методы для проверки представления “PersonalAccount”

По аналогии были написаны тесты для других форм, моделей и представлений.

## **Заключение**

В процессе работы были изучены средства разработки приложений на языке Python, произведено изучение и описание предметной области. была спроектирована архитектура системы, отвечающая всем требованиям, предъявленным к ней на этапе постановки задачи.

На основе анализа существующих на рынке аналогов, решающих нашу задачу, был спроектирован и реализован дружелюбный, интуитивно понятный пользовательский интерфейс.

При проектировании была учтена возможность расширения функционала системы и её масштабирования. Разработанный функционал протестирован модульными тестами.

## Список использованных источников

1. Python 3.8.3 documentation [Электронный ресурс] // Python 3.8.3 documentation. Режим доступа: <https://docs.python.org/3/>, свободный.
2. Bootstrap documentation [Электронный ресурс] // Bootstrap documentation. Режим доступа: <https://getbootstrap.com/docs/4.5/getting-started/introduction/>, свободный.
3. JSDoc [Электронный ресурс] // JSDoc. Режим доступа: <https://jsdoc.app>, свободный.
4. Рыбакова Г.Р., Кротова И.В., Камоза Т.Л., Моделирование совместимости учебной информации: методологические подходы [Текст] / Рыбакова Г.Р., Кротова И.В., Камоза Т.Л., 2015. — 11 с.
5. Ключков В.П., Зайцев А.В., Научные предпосылки возникновения междисциплинарного дихотомического подхода [Текст] / Ключков В.П., Зайцев А.В., 2013. — 4 с.
6. Шишкина Л.П., Мощанская Т.В., Логико-семантическая структура темы лексикона-тезауруса как основа структурирования предметно-тематического содержания текстов [Текст] / Шишкина Л.П., Мощанская Т.В., 2015. — 9 с.
7. Васильева Н.О., Современный системный подход и проблема совместимости текстов [Текст] / Васильева Н.О., 2012. — 7 с.
8. Ключков В.П., Васильева Н.О., Дихотомический анализ основной мерности сложности общенаучной категории “совместимость” [Текст] / Ключков В.П., Васильева Н.О., 2012. — 3 с.
9. Криони Н.К., Никин А.Д., Филиппова А.В. Автоматизированная система анализа сложности учебных текстов [Текст] / Криони Н.К., Никин А.Д., Филиппова А.В., 2008. — 7 с.
10. Оборнева И.В. Автоматизированная оценка сложности учебных текстов на основе статических параметров [Текст] / Оборнева И.В., 2006. — 20 с.
11. Влад Головач. Дизайн пользовательского интерфейса . [Электронный ресурс]. – Режим доступа: <http://en.bookfi.net/book/555670>, свободный.
12. Django and Ajax [Электронный ресурс] // Django and Ajax. Режим доступа: <https://kmv-it.ru/django-i-ajax>, свободный.

13. AJAX in Django with JQuery [Электронный ресурс] // AJAX in Django with JQuery. Режим доступа: <https://www.tangowithdjango.com/book17/chapters/ajax.html>, свободный.
14. Руководство часть 10: Тестирование приложений Django [Электронный ресурс] // Руководство по тестированию приложений Django. Режим доступа: <https://developer.mozilla.org/ru/docs/Learn/Server-side/Django/Testing>, свободный.
15. Widgets [Электронный ресурс] // Widgets. Режим доступа: <https://docs.djangoproject.com/en/3.0/ref/forms/widgets/>, свободный.
16. Base views [Электронный ресурс] // Base views. Режим доступа: <https://docs.djangoproject.com/en/3.0/ref/class-based-views/base/>, свободный.
17. Writing and running tests [Электронный ресурс] // Writing and running tests. Режим доступа: <https://docs.djangoproject.com/en/3.0/topics/testing/overview/>, свободный.
18. Unit testing framework [Электронный ресурс] // Unit testing framework. Режим доступа: <https://docs.python.org/3/library/unittest.html>, свободный.
19. unittest.TestCase Support [Электронный ресурс] // unittest.TestCase Support. Режим доступа: <https://pytest.org/en/latest/unittest.html>, свободный.
20. Django-registration [Электронный ресурс] // Django-registration. Режим доступа: <https://django-registration.readthedocs.io/en/3.1/>, свободный.
21. Custom user models [Электронный ресурс] // Custom user models. Режим доступа: <https://django-registration.readthedocs.io/en/3.1/custom-user.html>, свободный.
22. Using the Django authentication system [Электронный ресурс] // Using the Django authentication system. Режим доступа: <https://docs.djangoproject.com/en/3.0/topics/auth/default/>, свободный.
23. Models [Электронный ресурс] // Models. Режим доступа: <https://docs.djangoproject.com/en/3.0/topics/db/models/>, свободный.
24. Django учебник Часть 3: использование моделей [Электронный ресурс] // Использование моделей. Режим доступа:



<https://developer.mozilla.org/ru/docs/Learn/Server-side/Django/Models>,

свободный.

25. The Django Admin [Электронный ресурс] // The Django Admin. Режим доступа: <https://djangobook.com/mdj2-django-admin/>, свободный.

## Приложение А

### Задание на выполнение ВКР

Министерство науки и высшего образования Российской Федерации федеральное государственное бюджетное образовательное учреждение высшего образования «Алтайский государственный технический университет им. И. И. Ползунова»

#### УТВЕРЖДАЮ

Заведующий кафедрой ПМ

Кантор С. А.

*подпись*

*Ф.И.О*

#### ЗАДАНИЕ

на выполнение бакалаврской работы

*бакалаврской работы, дипломной работы (дипломного проекта), магистерской диссертации*

по направлению подготовки (специальности)

Программная инженерия

студенту группы ПИ-61 Оверченко Виолетте Романовне

*фамилия, имя, отчество*

**Тема:** Разработка web-интерфейса для анализа учебных материалов с учетом их внутренней и внешней совместимости

Утверждено приказом ректора от 13.02.20 № Л-321

Срок выполнения работы 15.06.20

Задание принял к исполнению В.Р. Оверченко

*подпись*

*инициалы, фамилия*

БАРНАУЛ 2020

## 1 Исходные данные

Задание на выполнение, техническая документация на языки программирования Python, на Django Framework, ресурсы internet

## 2 Содержание разделов ВКР и календарный график её выполнения

Наименование разделов работы и их содержание	Трудоемкость, %	Срок выполнения	Консультант (фамилия, инициалы, подпись)
1 Предметная область	30		Андреева А.Ю.
2 Проектирование	30		Андреева А.Ю.
3 Реализация	40		Андреева А.Ю.

## 3 Научно-библиографический поиск

3.1 По научно-технической литературе просмотреть РЖ «Кибернетика», «Программное обеспечение» и научно-технические журналы «Программирование», «Программная инженерия», «Информационные технологии»

за последние 10 лет.

3.2 По нормативной литературе просмотреть указатели государственных и отраслевых стандартов за последний год.

3.3 Патентный поиск провести за 4 года по странам Россия

Руководитель

подпись

А.Ю. Андреева

инициалы, фамилия

## Приложение Б

### Исходный код программы

Settings.py

```
import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'wlro!xg98srk26*a!alm3r+44hh1eur+csw5n*t#chopcp4sy!'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'modeltranslation',
    'analysis_material.apps.AnalysisMaterialConfig',
    'analyzer',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'django.contrib.sites',
    'allauth',
    'allauth.account',
    'allauth.socialaccount',
    'allauth.socialaccount.providers.vk',
    'crispy_forms',
    'widget_tweaks'
]
```

```

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.locale.LocaleMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'mysite2.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'templates')],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    ],
]

WSGI_APPLICATION = 'mysite2.wsgi.application'

# Database
# https://docs.djangoproject.com/en/3.0/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'analysis_material',
        'USER': 'root',
        'PASSWORD': '',
        'HOST': '127.0.0.1',
        'PORT': '3306',
        'OPTIONS': {
            'init_command': "SET sql_mode='STRICT_TRANS_TABLES'",

```

```

        'charset': 'utf8mb4',
    },
}
}

AUTHENTICATION_BACKENDS = (
    'django.contrib.auth.backends.ModelBackend',
    'allauth.account.auth_backends.AuthenticationBackend',
)
# Password validation
# https://docs.djangoproject.com/en/3.0/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

# Internationalization
# https://docs.djangoproject.com/en/3.0/topics/i18n/

LANGUAGE_CODE = 'ru-ru'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True

gettext = lambda s: s
LANGUAGES = (
    ('ru', gettext('Russia')),
    ('en', gettext('English')),

```

)

```
CRISPY_TEMPLATE_PACK = 'bootstrap4'
```

```
ACCOUNT_EMAIL_CONFIRMATION_EXPIRE_DAYS = 1
```

```
ACCOUNT_USERNAME_MIN_LENGTH = 4
```

```
LOGIN_REDIRECT_URL = '/'
```

```
EMAIL_BACKEND = 'django.core.mail.backends.dummy.EmailBackend'
```

```
# Static files (CSS, JavaScript, Images)
```

```
# https://docs.djangoproject.com/en/3.0/howto/static-files/
```

```
STATIC_URL = '/static/'
```

```
STATICFILES_DIRS = [  
    os.path.join(BASE_DIR, "static"),  
]
```

```
MEDIA_URL = 'media/'
```

```
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
```

```
SITE_ID = 1
```

urls.py

```
from django.contrib import admin  
from django.urls import include, path  
from django.conf import settings  
from django.conf.urls.static import static
```

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path("", include('analysis_material.urls')),  
    path('accounts/', include('allauth.urls')),  
]
```

```
if settings.DEBUG:
```

```
    urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

manage.py

```
import os
```

```

import sys

def main():
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'mysite2.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()

```

models.py

```

from django.db import models
from django.urls import reverse
from django.contrib.auth.models import User

class Document(models.Model):
    user = models.ForeignKey(User, verbose_name="Пользователь",
on_delete=models.SET_NULL, null=True, blank=True)
    name_document = models.CharField("Учебный материал", max_length=200)
    path_file = models.FileField("Файл", upload_to='documents/')
    uploaded_at = models.DateTimeField("Время загрузки", auto_now_add=True)

    def __str__(self):
        return self.name_document

    def get_absolute_url(self):
        return reverse("document_detail", kwargs={"pk": self.pk})

class Meta:
    verbose_name = "Учебный материал"
    verbose_name_plural = "Учебные материалы"

class AnalysisDocument(models.Model):
    compatibility_document = models.FloatField(default=0)

```



```
def __str__(self):
    return self.compatibility_document
```

```
class Relation_Document_Analysis(models.Model):
    document = models.ForeignKey(Document, on_delete=models.CASCADE)
    analysis_document = models.ForeignKey(AnalysisDocument,
on_delete=models.CASCADE)
```

```
class Keyword(models.Model):
    document = models.ForeignKey(Document, on_delete=models.CASCADE)
    description_keyword = models.CharField(max_length=100)
    periodicity_keyword = models.IntegerField(default=0)
```

views.py

```
from django.shortcuts import render, redirect
from django.views.generic import View, ListView, DetailView, CreateView, DeleteView
from analyzer.analyzer_class.read_file import ReadFile
from analyzer.analyzer_class.compatibility_analysis import CompatibilityAnalysis
from analyzer.analyzer_class.thesaurus import Thesaurus
from .models import Document, User
from django.urls import reverse_lazy
from .forms import DocumentForm
```

```
class HomeViews(View):
    """ Главная страница """
    def get(self, request):
        return render(request, "analysis_material/home.html")
```

```
class AnalysisMaterialViews(View):
    """ Страница для расчёта совместимости """
    def get(self, request):
        return render(request, "analysis_material/calculation_compatibility.html")
```

```
class PersonalAccountViews(ListView):
    """ Личный кабинет: список документов, список анализов """
    model = Document
    template_name = "analysis_material/personal_account.html"
    paginate_by = 10

    def get_queryset(self):
        return Document.objects.filter(user=self.request.user)
```

```
class DocumentDetailViews(DetailView):
```

```

""" Подробная информация о документе """
model = Document
template_name = 'analysis_material/document_detail.html'

class DocumentDeleteView(DeleteView):
    """ Удаление документа """
    model = Document
    template_name = 'analysis_material/personal_account.html'
    success_url = reverse_lazy('personal_account')

class DocumentCreateView(CreateView):
    """ Добавление документа """
    model = Document
    form_class = DocumentForm
    template_name = 'analysis_material/document_form.html'

    def form_valid(self, form):
        form.instance.user = self.request.user
        return super(DocumentCreateView, self).form_valid(form)

def result_analyze_one(request):
    # проверка чтобы либо текст либо файл
    # если файл не того формата тоже проверка
    name_document = ""
    one_text_area = ""
    path_file = None
    read_file = ReadFile()

    if 'name_document' in request.POST and request.POST['name_document']:
        name_document = request.POST['name_document']
    if 'one_text_area' in request.POST and request.POST['one_text_area']:
        one_text_area = request.POST['one_text_area']
    if 'path_file' in request.FILES and request.FILES['path_file']:
        path_file = request.FILES['path_file']
        path_file = read_file(path_file)

    thesaurus = Thesaurus()
    keys, tfidf = thesaurus(path_file)
    return render(request,
                  'analysis_material/results_one.html', context={'name_document':
name_document,
                                                                'keys': keys,
                                                                'tfidf': tfidf})

def result_analyze_two(request):

```

```

# проверка чтобы либо текст либо файл
# если файл не того формата тоже проверка
one_material_name = ""
two_material_name = ""
one_text_area = ""
two_text_area = ""
one_uploaded_file = None
two_uploaded_file = None
read_file = ReadFile()

if 'one_material_name' in request.POST and request.POST['one_material_name']:
    one_material_name = request.POST['one_material_name']
if 'one_text_area' in request.POST and request.POST['one_text_area']:
    one_text_area = request.POST['one_text_area']
if 'one_uploaded_file' in request.FILES and request.FILES['one_uploaded_file']:
    one_uploaded_file = request.FILES['one_uploaded_file']
    one_uploaded_file = read_file(one_uploaded_file)

if 'two_material_name' in request.POST and request.POST['two_material_name']:
    two_material_name = request.POST['two_material_name']
if 'two_text_area' in request.POST and request.POST['two_text_area']:
    two_text_area = request.POST['two_text_area']
if 'two_uploaded_file' in request.FILES and request.FILES['two_uploaded_file']:
    two_uploaded_file = request.FILES['two_uploaded_file']
    two_uploaded_file = read_file(two_uploaded_file)

compatibility_analysis = CompatibilityAnalysis()
one_data, two_data, result = compatibility_analysis(one_text_area, two_text_area,
                                                    one_uploaded_file, two_uploaded_file)

return render(request,
              'analysis_material/results_two.html', context={'one_material_name':
one_material_name,
                                                            'two_material_name': two_material_name,
                                                            'one_data': one_data,
                                                            'two_data': two_data,
                                                            'result': result})

```

urls.py

```

from django.urls import path
from . import views

```

```

urlpatterns = [
    path("", views.HomeViews.as_view(), name="home"), # главная страница

```

```

    path('calculation_compatibility/', views.AnalysisMaterialViews.as_view(),
name="calculation_compatibility"),
    path('result_analyze_two/', views.result_analyze_two, name="result_analyze_two"),
    path('result_analyze_one/', views.result_analyze_one, name="result_analyze_one"),
    path('personal_account/', views.PersonalAccountViews.as_view(),
name="personal_account"),
    path('<int:pk>/', views.DocumentDetailView.as_view(), name="document_detail"),
    path('delete/<int:pk>/', views.DocumentDeleteView.as_view(),
name='document_delete'),
    #path('delete/<int:pk>/', views.deleteDocument, name='deleteDocument'),
    path('new/', views.DocumentCreateView.as_view(), name='document_new'), # new
]

```

forms.py

```
from django import forms
```

```
from .models import Document
```

```

class DocumentForm(forms.ModelForm):
    class Meta:
        model = Document
        fields = ['name_document', 'path_file']

```

apps.py

```
from django.apps import AppConfig
```

```

class AnalysisMaterialConfig(AppConfig):
    name = 'analysis_material'

```

admin.py

```
from django.contrib import admin
```

```
from .models import Document, AnalysisDocument, Relation_Document_Analysis,
Keyword
```

```
admin.site.register(Document)
```

test\_forms.py

```

from django.test import TestCase
import datetime
from django.utils import timezone
from analysis_material.forms import DocumentForm

class DocumentFormTest(TestCase):

    def test_document_form_name_field_label(self):
        form = DocumentForm()
        self.assertTrue(form.fields['name_document'].label == None or
form.fields['name_document'].label == 'Учебный материал')

    def test_document_form_path_field_label(self):
        form = DocumentForm()
        self.assertTrue(form.fields['path_file'].label == None or form.fields['path_file'].label
== 'Файл')

```

test\_models.py

```

from django.test import TestCase

from analysis_material.models import Document

class DocumentModelTest(TestCase):

    @classmethod
    def setUpTestData(cls):
        Document.objects.create(name_document='История',
path_file='documents/1_2_главы.docx')

    def test_name_document_label(self):
        document = Document.objects.get(id=1)
        field_label = document._meta.get_field('name_document').verbose_name
        self.assertEqual(field_label, 'Учебный материал')

    def test_uploaded_at_label(self):
        document = Document.objects.get(id=1)
        field_label = document._meta.get_field('uploaded_at').verbose_name
        self.assertEqual(field_label, 'Время загрузки')

    def test_path_file_label(self):
        document = Document.objects.get(id=1)
        field_label = document._meta.get_field('path_file').verbose_name
        self.assertEqual(field_label, 'Файл')

```

```

def test_name_document_max_length(self):
    document = Document.objects.get(id=1)
    max_length = document._meta.get_field('name_document').max_length
    self.assertEqual(max_length, 200)

def test_object_name_is_name_document(self): # проверка для метода __str__
    document = Document.objects.get(id=1)
    expected_object_name = document.name_document
    self.assertEqual(expected_object_name, str(document))

def test_get_absolute_url(self):
    document = Document.objects.get(id=1)
    self.assertEqual(document.get_absolute_url(), '/1/')

```

test\_views.py

```

from django.test import TestCase
from analysis_material.models import Document
from django.urls import reverse

from django.contrib.auth.models import User # Необходимо для представления User
как borrower

class DocumentByUserListViewTest(TestCase):

    def setUp(self):
        # Создание двух пользователей
        test_user1 = User.objects.create_user(username='testuser1', password='12345')
        test_user1.save()
        test_user2 = User.objects.create_user(username='testuser2', password='12345')
        test_user2.save()

        number_of_documents = 13
        for document_num in range(number_of_documents):
            if document_num % 2:
                usertest = test_user1
            else:
                usertest = test_user2
            Document.objects.create(user = usertest, name_document = 'Материал %s' %
document_num, path_file='Файл %s' % document_num,)

    def test_logged_in_uses_correct_template(self):
        login = self.client.login(username='testuser1', password='12345')

```

```

resp = self.client.get(reverse('personal_account'))

# Проверка что пользователь залогинился
self.assertEqual(str(resp.context['user']), 'testuser1')
# Проверка ответа на запрос
self.assertEqual(resp.status_code, 200)

# Проверка того, что мы используем правильный шаблон
self.assertTemplateUsed(resp, 'analysis_material/personal_account.html')

#Подтверждение, что все документы принадлежат testuser1
for document in resp.context['document_list']:
    self.assertEqual(resp.context['user'], document.user)

```

base.html

```

<!doctype html>
{% load static %}
<html lang="en">
<head>
    <!-- Required meta tags -->
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">

    <!-- Bootstrap CSS -->

    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css"
integrity="sha384-
MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPM
O" crossorigin="anonymous">
    <link rel="shortcut icon" href="{ % static 'analysis_material/images/favicon.png' % }"
type="image/png">

    <link rel="stylesheet" type="text/css" href="{ % static
'analysis_material/css/home_style.css' % }">

    <title>Анализ учебных материалов</title>
</head>

<body >
    <nav class="navbar navbar-expand-lg navbar-dark bg-dark">
        <a class="navbar-brand" href="{ % url 'home' % }">

```

```

    { % load static % }
    
    Анализ учебных материалов
</a>
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-
expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarNavAltMarkup">
    <ul class="navbar-nav">
        <li class="nav-item">
            <a class="nav-link" href="{ % url 'document_new' % }">Анализ одного
материала</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="{ % url 'calculation_compatibility' % }">Расчёт
совместимости двух материалов</a>
        </li>
        { % if user.is_authenticated % }
        <li class="nav-item">
            <a class="nav-link" href="{ % url 'personal_account' % }">Личный кабинет</a>
        </li>
        { % endif % }
    </ul>
    <ul id="main-menu" class="navbar-nav ml-auto">
        { % if user.is_authenticated % }
        <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle" href="#" id="navbarDropdown"
role="button" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
                {{ user.get_username }}
            </a>
            <div class="dropdown-menu" aria-labelledby="navbarDropdown">
                <a class="dropdown-item" href="{ % url 'account_logout' % }">Выход</a>
            </div>
        </li>
        { % else % }
        <li class="nav-item">
            <a class="btn btn-secondary" id="btn_registration" href="{ % url 'account_signup'
% }" role="button">Регистрация</a>
        </li>
        <li class="nav-item">
            <a class="btn btn-secondary" href="{ % url 'account_login' % }"
role="button">Вход</a>

```



```

        </li>
    { % endif % }
</ul>
</div>
</nav>
{ % block content % }
{ % endblock % }
<footer class="footer">
    <div class="container">
        <span class="text-muted">Наши контакты: owerchenko.violetta@yandex.ru</span>
    </div>
</footer>
<!-- Optional JavaScript -->
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"
integrity="sha384-
OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI"
crossorigin="anonymous"></script>
<script type="text/javascript" src={ % static "analysis_material/js/analysis_material.js"
% }></script>
</body>
</html>

```

## Analysis\_two.html

```

{ % extends "base.html" % }
{ % block content % }

<section class="ftco-section ftco-no-pt">
    <form method="post" action="/result_analyze_two" enctype="multipart/form-data"
class="search-property" >
        <div class="container">
            <div class="row">
                <div class="text pb-4 pt-5">
                    <h1 class="mb-4">Анализ двух материалов</h1>
                    <div class="desc">

```

```

        <p>Данный анализ покажет насколько совпадают тезаурусы данных
материалов.</p>
        <p>Для добавления материалов введите название и текст в
соответствующие поля ниже.</p>
        <p>Для проведения анализа нажмите на кнопку "Выполнить анализ" в
самом низу страницы.</p>
    </div>
</div>
<div class="col-md-6">
    <div class="heading-section mb-4 ml-md-4 pl-md-5">
        <h3 class="mb-4">
            Первый материал для анализа
        </h3>
        <div class="row">
            <div class="col-md-12 align-items-end ftco-animate">
                <div class="form-group">
                    <label for="#">Название материала</label>
                    <div class="form-field">
                        <input type="text" class="form-control" placeholder="Введите
название"
                                name="one_material_name">
                    </div>
                </div>
            </div>
            <div class="col-md-12 align-items-end ftco-animate">
                <div class="form-group">
                    <label for="#">Текст материала</label>
                    <div class="form-field">
                        <textarea class="form-control" placeholder="Введите текст"
                                name="one_text_for_analyze"></textarea>
                    </div>
                    { % csrf_token % }
                    <input type="file" class="form-control"
name="one_uploaded_file"/>
                </div>
            </div>
        </div>
    </div>
</div>
<div class="col-md-6 d-flex">
    <div class="heading-section mb-4 ml-md-4 pl-md-5">
        <h3 class="mb-4">
            Второй материал для анализа
        </h3>

```

```

<div class="row">
  <div class="col-md-12 align-items-end ftco-animate">
    <div class="form-group">
      <label for="#">Название материала</label>
      <div class="form-field">
        <input type="text" class="form-control" placeholder="Введите
название"
        name="two_material_name">
      </div>
    </div>
  </div>
  <div class="col-md-12 align-items-end ftco-animate">
    <div class="form-group">
      <label for="#">Текст материала</label>
      <div class="form-field">
        <textarea class="form-control" placeholder="Введите текст"
        name="two_text_for_analyze"></textarea>
      </div>
      { % csrf_token % }
      <input type="file" class="form-control"
name="two_uploaded_file"/>
    </div>
  </div>
</div>

</div>
<div class="col-md-12">
  <div class="col-md-12 align-self-end ftco-animate">
    <div class="form-group">
      <div class="form-field">
        <input type="submit" value="Выполнить анализ"
        class="form-control btn btn-primary"/>
      </div>
    </div>
  </div>
</div>
</div>
</div>
</form>
</section>

<!-- loader -->
<div id="ftco-loader" class="show fullscreen">

```

```

<svg class="circular" width="48px" height="48px">
  <circle class="path-bg" cx="24" cy="24" r="22" fill="none" stroke-width="4"
stroke="#eeeeee"/>
  <circle class="path" cx="24" cy="24" r="22" fill="none" stroke-width="4" stroke-
miterlimit="10"
stroke="#F96D00"/>
</svg>
</div>

{ % endblock % }

```

Calculation\_compability.html

```

{ % extends "base.html" % }

{ % block content % }

<div class="jumbotron">
  <div class="container">
    <h1 class="display-3">Инструкция</h1>
    <p>Возможно, здесь стоит описать порядок действий для анализа.</p>
  </div>
</div>

<form method="post" action="{ % url 'result_analyze_two' % }" enctype="multipart/form-
data" class="search-property" >
  <div class="container">
    <div class="row">

      <div class="col-sm-6">
        <div class="container border border-secondary rounded mycol-sm-6">
          { % csrf_token % }
          <div class="form-group">
            <label for="nameDocument"><h4>Введите название 1 учебного
материала:</h4></label>
            <input type="text" class="form-control" id="nameDocument"
name="one_material_name" required>
          </div>
          <div class="form-group" >
            <label for="exampleFormControlFile1">Прикрепите файл для анализа (дох,
docx):</label>

```

```

        <input type="file" class="form-control-file" id="exampleFormControlFile1"
name="one_uploaded_file">
    </div>
    <textarea class="form-control" id="exampleFormControlTextarea1"
placeholder="Вставьте текст для анализа" rows="15" name="one_text_for_analyze"
></textarea><br>
    </div>
</div>
<div class="col-sm-6 ">
    <div class="container border border-secondary rounded mycol-sm-6">
        <div class="form-group">
            <label for="nameDocument"><h4>Введите название 2 учебного
материала:</h4></label>
            <input type="text" class="form-control" id="nameDocument"
name="two_material_name" required>
            </div>
            <div class="form-group" >
                <label for="exampleFormControlFile1">Прикрепите файл для анализа (docx,
docx):</label>
                <input type="file" class="form-control-file" id="exampleFormControlFile1"
name="two_uploaded_file">
                </div>
                <textarea class="form-control" id="exampleFormControlTextarea1"
placeholder="Вставьте текст для анализа" rows="15" name="one_text_for_analyze"
></textarea><br>
            </div>
        </div>
        <div class="col-sm text-center">
            <input type="submit" value="Рассчитать совместимость" class="form-control btn
btn-secondary btn-block"/>
        </div>
    </div>
</div>
</form>

{% endblock %}

```

Document\_detail.html

```

{% extends "base.html" %}

{% block content %}
    {% load static %}
    <div class="container">

```

```

<h1>{{ document.name_document }}</h1>
<div class="text-right">
  <form method="post" action="{ % url 'document_delete' document.pk % }">
    { % csrf_token % }
    <button type="submit">
      
    </button>
  </form>
</div>
<table class="table">
  <thead class="thead-dark">
    <tr>
      <th scope="col">Ключевое слово</th>
      <th scope="col">Частота</th>
    </tr>
  </thead>
  <tbody>
    { % for keyword in document.keyword_set.all % }
    <tr>
      <td>{{ keyword.description_keyword }}</td>
      <td>{{ keyword.periodicity_keyword }}</td>
    </tr>
    { % endfor % }
  </tbody>
</table>
</div>

{ % endblock % }

```

## Document\_form.html

```

{ % extends 'base.html' % }
{ % load crispy_forms_tags % }

{ % block content % }
<div class="container">
  <h1>Добавление учебного материала:</h1>
  <form action="{ % url 'result_analyze_one' % }" method="post" enctype
="multipart/form-data">
    { % csrf_token % }

    { % if user.is_authenticated % }
    <div class="form-group">

```

```

        {{ form.name_document|as_crispy_field }}
    </div>
    {% endif %}
    <div class="form-group">
        <label for="exampleFormControlFile1">Прикрепите файл для анализа (docx, docx):</label>
        {{ form.path_file|as_crispy_field }}
    </div>
    <textarea class="form-control" id="exampleFormControlTextarea1"
placeholder="Вставьте текст для анализа" rows="15" name="one_text_for_analyze"
></textarea><br>
    <input type="submit" value="Добавить" class="form-control btn btn-secondary btn-block"/>
    </form>
</div>

{% endblock content %}

```

## Home.html

```

{% extends "base.html" %}

{% block content %}
    {% load static %}
    <div class="container text-center">
        <main role="main" class="inner cover">
            <h1 class="cover-heading">Анализ совместимости учебных материалов</h1>
            <p class="lead">Здесь должен быть какой-то красивый текст, описывающий функциональность этого сервиса!</p>
            <p class="lead">
                <a href="{% url 'calculation_compatibility' %}" class="btn btn-lg btn-secondary">Рассчитать совместимость</a>
            </p>
        </main>
    </div>

{% endblock %}

```

## Personal\_account.html

```

{% extends "base.html" %}
{% load static %}

```

```

{% block content %}
<div class="container">
  <h1>Личный кабинет: {{ user.get_username }}</h1>
  <ul class="nav nav-tabs" id="tab-example" role="tablist">
    <li class="nav-item">
      <a class="nav-link active" id="first-tab" data-toggle="tab"
        href="#first-tab-content" role="tab" aria-controls="first-tab-content"
        aria-selected="true">Загруженные файлы</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" id="second-tab" data-toggle="tab"
        href="#second-tab-content" role="tab" aria-controls="second-tab-content"
        aria-selected="false">Результаты анализа</a>
    </li>
  </ul>

  <div class="tab-content" id="tab-content-example">
    <div class="tab-pane fade show active" id="first-tab-content" role="tabpanel"
      aria-labelledby="first-tab">
      {% if document_list %}
      <div class="table-wrapper-scroll-y my-custom-scrollbar">
        <table class="table table-bordered table-striped table-hover">
          <tbody>
            {% for document in document_list %}
            <tr>
              <td><a href="{{ document.get_absolute_url
}}">{{ document.name_document }}</a></td>
              <td width="15px">
                <form method="post" action="{% url 'document_delete' document.pk
%}">
                  {% csrf_token %}
                  <button type="submit">
                    
                  </button>
                </form>
              </td>
            </tr>
            {% endfor %}
          </tbody>
        </table>
      </div>
      <p>Для загрузки файлов для анализа нажмите
      <a href="{% url 'document_new' %}">сюда.</a></p>
      {% else %}

```



```

    <p>Для загрузки файлов для анализа нажмите
    <a href="{ % url 'document_new' % }">сюда.</a></p>
    { % endif % }
</div>
<div class="tab-pane fade" id="second-tab-content" role="tabpanel"
    aria-labelledby="second-tab">
    { % if document_list % }
    <div class="table-wrapper-scroll-y my-custom-scrollbar">
    <table class="table table-bordered table-striped table-hover">
    <thead>
    <tr>
    <th scope="col">Первый материал</th>
    <th scope="col">Второй материал</th>
    <th scope="col">Совместимость</th>
    <th scope="col"></th>
    </tr>
    </thead>
    <tbody>
    { % for document in document_list % }
    <tr>
    <td>{ { document.name_document } }</td>
    <td>{ { document.name_document } }</td>
    <td>75%</td>
    <td width="15px">
    
    </td>
    </tr>
    { % endfor % }
    </tbody>
    </table>
    </div>
    <p>Для проведения анализа нажмите
    <a href="{ % url 'calculation_compatibility' % }">сюда.</a></p>
    { % else % }
    <p>Для проведения анализа нажмите
    <a href="{ % url 'calculation_compatibility' % }">сюда.</a></p>
    { % endif % }
</div>
</div>
</div>

{ % endblock % }

```

## Results\_one.html

```
{% include "base.html" %}
{% load static %}
{% block content %}

<div class="container">
  <h1>Результаты анализа</h1>
  <h5>Наименование материала: {{ name_document }}</h5>
  <h5>Тезаурус</h5>
  <form method="GET" action="{% url 'personal_account' %}" enctype="multipart/form-
data" class="search-property" >
    {% csrf_token %}
    <input type="submit" value="Добавить в библиотеку" class="form-control btn btn-
primary"/>
    <div class="table-wrapper-scroll-y my-custom-scrollbar">
      <table class="table table-bordered table-striped table-hover">
        <thead>
          <tr>
            <th scope="col"></th>
            <th scope="col">Ключевое слово</th>
          </tr>
        </thead>
        <tbody>
          {% for key in keys %}
            <tr>
              <td width="10px">
                <div class="custom-control custom-checkbox">
                  <input type="checkbox" class="custom-control-input" id="customCheck"
checked>
                  <label class="custom-control-label" for="customCheck"></label>
                </div>
              </td>
              <td>{{ key }}</td>
            </tr>
          {% endfor %}
        </tbody>
      </table>
    </div>
  </form>

</div>
```

```
{% endblock % }
```

## Results\_two.html

```
{% include "base.html" % }
```

```
{% load static % }
```

```
{% block content % }
```

```
<section class="ftco-section ftco-no-pt">
```

```
  <div class="container">
```

```
    <div class="row">
```

```
      <div class="text pb-4 pt-5">
```

```
        <h1 class="mb-4">Результаты анализа</h1>
```

```
        <div class="desc">
```

```
          <p>Данный анализ показывает насколько совпадают тезауусы данных  
материалов.</p>
```

```
          <h4 class="mb-4">Совпадение: {{ result }}%</h4>
```

```
        </div>
```

```
      </div>
```

```
      <div class="col-md-6">
```

```
        <div class="heading-section mb-4 ml-md-4 pl-md-5">
```

```
          <h5 class="mb-4">Наименование материала: {{ one_material_name  
}}</h5>
```

```
          <h5 class="mb-4">
```

```
            Тезаурус
```

```
          </h5>
```

```
          <table class="table table-bordered table-hover">
```

```
            <thead class="thead-dark">
```

```
              <tr>
```

```
                <th scope="col"></th>
```

```
                <th scope="col">Ключевое слово</th>
```

```
              </tr>
```

```
            </thead>
```

```
            <tbody>
```

```
              {% for key, value in one_data % }
```

```
                <tr>
```

```
                  <th scope="row" class="table-fragment"><input type="checkbox" checked></th>
```

```
                  <td class="table-fragment">{{ key }}</td>
```

```
                </tr>
```

```
              {% endfor % }
```

```

        </tbody>
    </table>
</div>
</div>
<div class="col-md-6 ">
    <div class="heading-section mb-4 ml-md-4 pl-md-5">
        <h5 class="mb-4">Наименование материала: {{ two_material_name
    }}</h5>
        <h5 class="mb-4">
            Тезаурус
        </h5>
        <table class="table table-bordered table-hover">
            <thead class="thead-dark">
                <tr>
                    <th scope="col"></th>
                    <th scope="col">Ключевое слово</th>
                </tr>
            </thead>
            <tbody>
                {% for key, value in two_data %}
                <tr>
                    <th scope="row" class="table-fragment"><input type="checkbox"
checked></th>
                    <td class="table-fragment">{{ key }}</td>
                </tr>
                {% endfor %}
            </tbody>
        </table>
    </div>
</div>
</div>
</div>
</section>

```

```

<!-- loader -->
<div id="ftco-loader" class="show fullscreen">
    <svg class="circular" width="48px" height="48px">
        <circle class="path-bg" cx="24" cy="24" r="22" fill="none" stroke-width="4"
stroke="#eeeeee"/>
        <circle class="path" cx="24" cy="24" r="22" fill="none" stroke-width="4" stroke-
miterlimit="10"
stroke="#F96D00"/>
    </svg>
</div>

```

```
{% endblock %}
```

## Login.html

```
{% extends "base.html" %}
```

```
{% load i18n %}
```

```
{% load account socialaccount %}
```

```
{% block content %}
```

```
<h1 align="center">Авторизация</h1>
```

```
<form class="login form-signin text-center" method="post" action="{% url  
'account_login' %}">
```

```
    {% csrf_token %}
```

```
    {% load static %}
```

```
    
```

```
    <label for="inputUsername" class="sr-only">Логин</label>
```

```
    <input type="text" id="inputUsername" class="form-control" placeholder="Логин"  
name="login" required autofocus>
```

```
    <label for="inputPassword" class="sr-only">Пароль</label>
```

```
    <input type="password" id="inputPassword" class="form-control"  
placeholder="Пароль" name="password" required>
```

```
    {% if redirect_field_value %}
```

```
    <input type="hidden" name="{{ redirect_field_name }}" value="{{ redirect_field_value  
}}"/>
```

```
    {% endif %}
```

```
    <input class="btn btn-lg btn-secondary btn-block" type="submit" value="Войти">
```

```
    <a href="{% url 'account_reset_password' %}">Забыли пароль?</a><br>
```

```
    <a href="{{ signup_url }}">Создать аккаунт</a>
```

```
</form>
```

```
{% endblock %}
```

## Logout.html

```
{% extends "account/base.html" %}
```

```
{% load i18n %}
```

```
{% block head_title %}{% trans "Sign Out" %}{% endblock %}
```

```

{% block content %}
<div class="container text-center">
  <h1 align="center">{% trans "Sign Out" %}</h1>

  <p>{% trans 'Are you sure you want to sign out?' %}</p>

  <form method="post" action="{% url 'account_logout' %}">
    {% csrf_token %}
    {% if redirect_field_value %}
    <input type="hidden" name="{% redirect_field_name %}" value="{%
redirect_field_value %}" />
    {% endif %}
    <button class="btn btn-lg btn-secondary btn-block" type="submit">{% trans 'Sign Out'
%}</button>
  </form>
</div>

{% endblock %}

```

## Signup.html

```

{% extends "account/base.html" %}

{% load i18n %}

{% block head_title %}{% trans "Signup" %}{% endblock %}

{% block content %}

<h1 align="center">Регистрация</h1>
<form class="form-signin text-center" method="post" action="{% url 'account_signup'
%}">
  {% csrf_token %}
  {% load static %}
  
  <label for="inputUsername" class="sr-only">Логин</label>
  <input type="text" id="inputUsername" class="form-control" placeholder="Логин"
name="username" required autofocus>
  <label for="inputEmail" class="sr-only">Email</label>
  <input type="email" id="inputEmail" class="form-control" placeholder="Email"
name="email" required>
  <label for="inputPassword1" class="sr-only">Пароль</label>

```

```

<input type="password" id="inputPassword1" class="form-control"
placeholder="Пароль" name="password1" required>
<label for="inputPassword2" class="sr-only">Повторите пароль</label>
<input type="password" id="inputPassword2" class="form-control"
placeholder="Повторите пароль" name="password2" required>
{ % if redirect_field_value % }
    <input type="hidden" name="{{ redirect_field_name }}" value="{{
redirect_field_value }}" />
{ % endif % }
<input class="btn btn-lg btn-secondary btn-block" type="submit"
value="Регистрация">
<a href="{{ login_url }}">Уже есть аккаунт?</a>
</form>
{ % endblock % }

```

### Signup\_closed.html

```

{ % extends "account/base.html" % }

{ % load i18n % }

{ % block head_title % }{ % trans "Sign Up Closed" % }{ % endblock % }

{ % block content % }
<h1>{ % trans "Sign Up Closed" % }</h1>

<p>{ % trans "We are sorry, but the sign up is currently closed." % }</p>
{ % endblock % }

```

### Email\_confirm.html

```

{ % extends "account/base.html" % }

{ % load i18n % }
{ % load account % }

{ % block head_title % }{ % trans "Confirm E-mail Address" % }{ % endblock % }

{ % block content % }
<h1>{ % trans "Confirm E-mail Address" % }</h1>

{ % if confirmation % }

```

```
{% user_display confirmation.email_address.user as user_display %}

<p>{% blocktrans with confirmation.email_address.email as email %}Please confirm that
<a href="mailto:{{ email }}">{{ email }}</a> is an e-mail address for user {{
user_display }}.{% endblocktrans %}</p>

<form method="post" action="{% url 'account_confirm_email' confirmation.key %}">
{% csrf_token %}
    <button type="submit">{% trans 'Confirm' %}</button>
</form>

{% else %}

{% url 'account_email' as email_url %}

<p>{% blocktrans %}This e-mail confirmation link expired or is invalid. Please <a
href="{{ email_url }}">issue a new e-mail confirmation request</a>.{% endblocktrans
%}</p>

{% endif %}

{% endblock %}
```

Home\_style.css

```
html {
    position: relative;
    min-height: 100%;
}
body {
    /* Нижний край поля по высоте нижнего колонтитула (высота футера)*/
    margin-bottom: 60px;
    background-color: #f7f7f7;
}

.footer {
    position: absolute;
    bottom: 0;
    width: 100%;
    /* Устанавливаем фиксированную высоту футера */
    height: 60px;
    line-height: 60px; /* Вертикально центрируем текст футера */
    background-color: #333333;
    text-align: center;
}
```



```

h1 {
  padding: 3rem 1.5rem;
}

.nav-item {
  margin-right: 10px;
}

.table {
  margin-top: 50px;
}

/*
 * Cover
 */
.cover {
  padding: 0 1.5rem;
}
.cover .btn-lg {
  padding: .75rem 1.25rem;
  font-weight: 700;
}

.mycol-sm-6 {
  padding: 20px;
}

.form-signin {
  width: 100%;
  max-width: 330px;
  padding-top: 10px;
  margin: auto;
}
.form-signin{
  font-weight: 400;
}
.form-signin .form-control {
  position: relative;
  box-sizing: border-box;
  height: auto;
  padding: 10px;
  font-size: 16px;
}
.form-signin .form-control:focus {

```

```

    z-index: 2;
}
input[type="submit"] {
    margin-top: 10px;
}

p#first {
    cursor: pointer;
    line-height: 13px;
    text-indent: 22px;
    line-height: 33px;
    border: 1px solid #d2d2d2;
    font-size: 18px;
}

p#first_yelloy {
    cursor: pointer;
    font-size: 18px;
    text-indent: 22px;
    line-height: 33px;
    border: 1PX SOLID #d2d2d2;
}

p#first2 {
    cursor: pointer;
    line-height: 13px;
    text-indent: 22px;
    line-height: 33px;
    border: 1px solid #d2d2d2;
    font-size: 18px;
}

p#first_yelloy2 {
    cursor: pointer;
    font-size: 18px;
    text-indent: 22px;
    line-height: 33px;
    border: 1PX SOLID #d2d2d2;
}

#result {
    margin-top: 20px;
    margin-bottom: 50px;
}

```

Analysis\_material.js

```
function first() {  
document.getElementById("second_hide").setAttribute("style", "opacity:1; transition: 1s;  
height: 100%;");  
document.getElementById("first").setAttribute("style", "display: none");  
document.getElementById("first_yelloy").setAttribute("style", "display: block");  
}
```

```
function first_yelloy() {  
document.getElementById("second_hide").setAttribute("style", "display: none");  
document.getElementById("first_yelloy").setAttribute("style", "display: none");  
document.getElementById("first").setAttribute("style", "display: block");  
}
```

```
function first2() {  
document.getElementById("second_hide2").setAttribute("style", "opacity:1; transition: 1s;  
height: 100%;");  
document.getElementById("first2").setAttribute("style", "display: none");  
document.getElementById("first_yelloy2").setAttribute("style", "display: block");  
}
```

```
function first_yelloy2() {  
document.getElementById("second_hide2").setAttribute("style", "display: none");  
document.getElementById("first_yelloy2").setAttribute("style", "display: none");  
document.getElementById("first2").setAttribute("style", "display: block");  
}
```