

**Министерство науки и высшего образования Российской Федерации
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»
(Университет ИТМО)**

Факультет **Прикладной информатики**

Направление подготовки **45.03.04 Интеллектуальные системы в гуманитарной
сфере**

Образовательная программа **Языковые модели и искусственный интеллект**

КУРСОВОЙ ПРОЕКТ

Тема: «Прототип платформы для адаптации текстов русскоязычных учебных
материалов для иностранных студентов ‘EduAdapt’»

Обучающийся: Бондар Илона Александровна, К3162

Санкт-Петербург 2024

СОДЕРЖАНИЕ

1 ВВЕДЕНИЕ.....	3
1.1 Актуальность темы.....	3
1.2 Цель проекта.....	5
1.3 Задачи проекта.....	5
2 ОСНОВНАЯ ЧАСТЬ.....	8
2.1 Описание работы над проектом.....	8
2.2 Анализ работы.....	15
2.3 Взаимодействие с командой.....	16
2.4 Оценка работы руководителя.....	16
3 ЗАКЛЮЧЕНИЕ.....	17
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	19
ПРИЛОЖЕНИЕ. ТЕХНИЧЕСКОЕ ЗАДАНИЕ К ПРОЕКТУ.....	20

1 ВВЕДЕНИЕ

1.1 Актуальность темы

Вопрос эффективной адаптации иностранных студентов играет важную роль для развития конкурентоспособности российской системы образования конкуренции на международном рынке образования. В условиях глобализации и растущей интеграции мировых образовательных процессов, задачи интернационализации университетов и привлечения иностранных студентов становятся стратегически важными. Высшие учебные заведения России активно работают над укреплением своих позиций на международной арене, и обеспечение качественной адаптации студентов из других стран является одной из ключевых составляющих этого процесса. Глобализация и межкультурное взаимодействие становятся все более существенными, университеты по всему миру сталкиваются с необходимостью эффективной адаптации своих учебных материалов для студентов из разных стран.

Однако, несмотря на все усилия, множество иностранных студентов сталкивается с трудностями, которые затрудняют их интеграцию в образовательную среду. Среди таких проблем можно выделить низкий уровень языковой компетенции, сложности в восприятии сложных научных текстов и терминов, а также непривычные методики обучения, которые могут отличаться от стандартов, к которым студенты привыкли на родине.

Эффективная адаптация учебных материалов для иностранных студентов в университете представляет собой сложный и многогранный процесс, который требует учета индивидуальных потребностей, уровня языковой компетенции и культурного контекста студентов. Процесс адаптации включает несколько ключевых аспектов: упрощение языковых конструкций и терминологии, предоставление дополнительных пояснений к сложным

понятиям, а также создание условий для интерактивного изучения материала. Более того, важно учитывать культурные особенности и менталитет студентов, что способствует более комфортной интеграции в новую образовательную среду.

Эта проблема становится более насущной в свете растущего числа студентов, выбирающих обучение за пределами своей родной страны.

Актуальность проблемы адаптации учебных материалов для иностранных студентов в университете подчеркивается не только повышением мобильности студентов и их стремлением к получению высшего образования в международном контексте, но и глубоким влиянием, которое этот процесс оказывает на качество образования в целом. Данный вопрос также связан с репутацией образовательных учреждений на международной арене. Университеты, которые успешно справляются с адаптацией иностранных студентов, становятся более привлекательными для абитуриентов из-за рубежа, что способствует укреплению их позиций в мировых рейтингах. Таким образом, адаптация учебных материалов — это не только способ улучшения образовательного процесса, но и важный элемент стратегии развития университета.

Иностранные студенты часто сталкиваются с языковыми и культурными барьерами, которые могут затруднить их академическую успешность и социокультурную адаптацию. В целях улучшения адаптации иностранных студентов к учебному процессу в университете, возможным решением является создание специализированной платформы, направленной на адаптацию учебных материалов. Современные технологии обработки естественного языка, машинного обучения и интерфейсы, ориентированные на пользователя, открывают новые возможности для создания таких платформ. Например, автоматическое упрощение текста с учетом лексического уровня

студента и предоставление интерактивных инструментов для изучения слов и терминов значительно облегчают процесс адаптации.

Такая платформа может предоставлять учебные ресурсы, адаптированные к уровню языковой подготовки студентов, а также учитывать их культурные особенности. Разработка такой платформы позволит адаптировать учебные тексты, учебные задания и материалы, делая их более доступными и понятными для студентов, для которых языковой барьер представляет значительное препятствие. Кроме того, использование подобной платформы может быть полезным не только для иностранных студентов, но и для преподавателей. Она позволит преподавателям создавать материалы, которые будут лучше восприниматься их аудиторией, а также предоставит инструменты для мониторинга прогресса студентов и корректировки учебного процесса.

1.2 Цель проекта

Цель проекта - Создать прототип платформы, которая упрощает образовательные материалы для иностранных студентов, обеспечивая удобный доступ к адаптированным текстам и интерактивным инструментам, таким как словарь незнакомых слов и упрощение учебного материала.

1.3 Задачи проекта

В рамках исследования для меня были сформулированы несколько ключевых задач. Эти задачи определяют направления дальнейшей работы и включают проектирование модулей регистрации и авторизации, разработку

функциональности словаря, а также создание временного API для обработки текста.

Одной из базовых задач является создание модуля регистрации и авторизации. Этот модуль играет центральную роль в персонализации доступа пользователей к системе. Регистрация должна предусматривать ввод логина, пароля и роли пользователя, где роли (например, студент или преподаватель) определяют уровень доступа к функционалу системы. Авторизация, в свою очередь, обеспечит проверку введенных данных (логин и пароль) для подтверждения подлинности пользователя.

Следующим этапом является реализация словаря, который будет выполнять образовательную функцию. Словарь предусматривает возможность добавления слов преподавателями с сопроводительными комментариями, что позволяет пояснить значения или контекст использования слов. Студенты, в свою очередь, могут просматривать добавленные слова. Для реализации этого функционала потребуется создание интуитивно понятного интерфейса, а также структуры данных, которая позволит эффективно хранить и отображать слова и комментарии. Особое внимание будет уделено разграничению прав доступа: только преподаватели смогут добавлять новые элементы, а студенты — просматривать их.

Дополнительно рассматривается задача разработки API, которое будет выполнять функцию промежуточного решения для обработки текста. Данное API должно принимать текстовые данные и возвращать их в неизменённом виде. Это решение служит временной мерой до внедрения алгоритма упрощения текста, который будет предназначен для адаптации текстов под различные уровни понимания целевой аудитории. Важной частью реализации данного API является обеспечение его гибкости и расширяемости, чтобы впоследствии можно было интегрировать более сложный функционал.

Рассматриваемые задачи закладывают основу для построения функциональной, удобной и безопасной системы, которая будет удовлетворять потребности преподавателей и иностранных студентов. Реализация перечисленных модулей требует последовательного подхода, включающего разработку архитектуры, выбор технологий и проведение тестирования. Эти аспекты станут предметом дальнейших этапов работы.

2 ОСНОВНАЯ ЧАСТЬ

2.1 Описание работы над проектом

Проект «EduAdapt» представляет собой веб-платформу, ориентированную на помощь иностранным студентам в изучении русскоязычных учебных материалов. Основная цель платформы — облегчить понимание сложных текстов за счет их автоматической адаптации и предоставить пользователям удобные инструменты для работы с лексикой. Проблема языкового барьера является одной из ключевых для иностранных студентов, обучающихся в русскоязычной среде, особенно когда учебные материалы насыщены научной терминологией, сложными конструкциями и идиомами, а преподаватели на лекциях говорят очень быстро и непонятно.

Руководитель моего проекта совместно с К. Третьяковой, Р. Кукряковой и А. Луцкович провел опрос среди иностранных студентов, целью которого стало изучение трудностей, с которыми они сталкиваются во время учебного процесса. Среди респондентов наиболее частыми проблемами оказались трудности с формулировкой вопросов и страх обратиться к преподавателю. Также студенты отмечали чрезмерную скорость речи преподавателей, что затрудняло восприятие материала. Только два студента указали на сложность содержания учебных материалов, а один респондент жаловался на нежелание преподавателя устанавливать контакт (Рисунок 1).

26 atsakymai

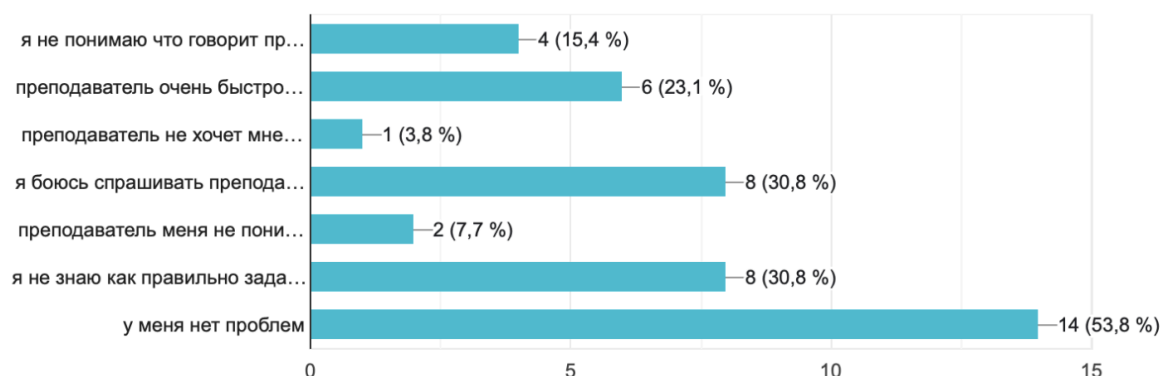


Рисунок 1 - Диаграмма опроса иностранных студентов

Для решения этой задачи, было принято решение создать платформу, которая могла бы предлагает инструменты, которые способствуют не только упрощению текстов, но и расширению лексического запаса. Основные модули системы включают:

- Функцию упрощения текста, которая на основе технологий обработки естественного языка преобразует сложные предложения в более понятные;
- Словарь, где преподаватели могут добавлять слова и комментарии, а студенты просматривать их в адаптированном формате;
- Систему регистрации и авторизации, которая разграничивает права пользователей в зависимости от их роли (преподаватель или студент).

Проект был выполнен в рамках командной работы с участием Грачевой С. А., Просветовой В. Д., Сармусуковой А. С., Цзян Ю., где каждому участнику были распределены индивидуальные задачи. Я взяла на себя роль бэкэнд-разработчика. В мои обязанности входила работа над серверной частью проекта, взаимодействие с базой данных и API для фронтенда. Я отвечала за разработку модуля регистрации и авторизации, создание словаря и реализацию API для обработки текста.

Работа над проектом проходила в несколько этапов:

— Постановка целей и задач:

На начальном этапе работы над проектом было проведено онлайн совещание, на котором Филипс более подробно рассказал об основной проблеме, которую решает проект, — это языковая адаптация материалов для иностранных студентов. Вместе с руководителем были определены технологии, которые будут использоваться для реализации, например, Python для обработки текста, Figma для проектирования интерфейсов и Google Colab для тестирования алгоритмов. Кроме того, была сформулирована общая цель совместной работы и декомпозированы задачи. Были получены следующие задачи:

1. Реализовать модуль регистрации и авторизации;
2. Создать функционал словаря;
3. Разработать временный API, который принимает текст и возвращает его без изменений.

Этап проектирования и разработки

В рамках реализации проекта были разработаны несколько ключевых модулей. Для их создания использовалась стандартная архитектура Django. Django — это высокоуровневый веб-фреймворк на языке Python, который обеспечивает удобные инструменты для создания полнофункциональных веб-приложений. Его архитектура основана на концепции разделения логики, данных и представления интерфейса, что делает код структурированным и легко расширяемым. Стандартная архитектура Django опирается на шаблон Model-View-Template (MVT), где каждая составляющая имеет свою четко определенную роль. Рассмотрим, как эта архитектура применяется для создания модуля регистрации и авторизации, реализации словаря и API, принимающего текст и возвращающего его.

Прежде всего, в архитектуре Django центральным элементом является модель данных (Model), которая отвечает за взаимодействие с базой данных. Она описывает структуру хранимых данных, их типы и связи между

объектами. Используя объектно-реляционное отображение (ORM), модели Django обеспечивают простой и интуитивный способ работы с данными без необходимости написания сложных SQL-запросов. Логика работы приложения сосредоточена в представлениях (View), которые принимают запросы от пользователя, обрабатывают их, взаимодействуют с моделями и возвращают соответствующий ответ. Наконец, шаблоны (Template) обеспечивают создание динамического интерфейса, предоставляя удобный инструмент для отображения данных, полученных из представлений.

Для реализации модуля регистрации и авторизации Django предоставляет встроенные средства, которые минимизируют усилия разработчика. Основой является модуль ``django.contrib.auth``, предоставляющий готовую модель пользователя с возможностью ее настройки. Регистрация пользователя включает создание записи в базе данных с проверкой корректности введенных данных. Авторизация, в свою очередь, предполагает процесс проверки личности пользователя с помощью аутентификации. Для этого Django использует систему форм, защищенных от CSRF-атак, а также механизмы хеширования паролей для их безопасного хранения. Все взаимодействия с модулем регистрации и авторизации происходят через четко определенные URL-адреса, которые связывают запросы с соответствующими представлениями. Это позволяет гибко настраивать функционал и интегрировать его с другими частями приложения.

Создание словаря в Django предполагает проектирование структуры данных, способной эффективно хранить термины и их определения. Такая структура описывается через модель, в которой каждому термину соответствует текстовое поле для определения. Приложение словаря может включать возможность добавления, удаления и редактирования записей, что реализуется через стандартные CRUD-операции. Логика обработки запросов сосредоточена в представлениях, где определяется, как пользователь взаимодействует со словарем, будь то просмотр списка терминов или работа с

отдельными записями. Шаблоны, в свою очередь, обеспечивают отображение этих данных в удобной и интуитивно понятной форме, применяя стилизованные HTML-страницы. Таким образом, использование стандартной архитектуры Django позволяет организовать работу словаря так, чтобы обеспечить высокую производительность и удобство работы как для пользователя, так и для разработчика.

Реализация API, принимающего текст и возвращающего его, в контексте Django также основывается на принципах MVT-архитектуры. Здесь модели могут не использоваться, так как API может работать исключительно с данными, переданными в запросе. Представления, однако, играют ключевую роль, обрабатывая входящие запросы и формируя ответы. API взаимодействует с клиентом через стандартные протоколы, такие как HTTP, а данные передаются в формате JSON. Для реализации подобных интерфейсов в Django используется встроенный функционал, который можно дополнительно улучшить с помощью сторонних инструментов, таких как Django REST Framework. Этот фреймворк упрощает создание API, предоставляя средства для сериализации данных и обработки запросов. В данном случае текст, переданный через API, обрабатывается сервером, после чего возвращается клиенту в виде ответа. Такой подход делает приложение универсальным и легко интегрируемым с другими системами. Был реализован модуль регистрации и авторизации, основываясь на возможностях Django и Django REST Framework.

Для регистрации пользователей была создана модель (Рисунок 2), которая расширяет базовую модель Django, добавив поле для роли пользователя. Процесс регистрации был организован через API: пользователи вводят логин, пароль и указывают свою роль. Для авторизации я использовала JWT-токены (Рисунок 3), которые обеспечивают безопасное хранение данных и возможность повторного доступа к системе без необходимости вводить пароль заново. Также была внедрена система распределения прав доступа,

чтобы преподаватели могли добавлять учебные материалы, а студенты — только просматривать их.

```
from django.contrib.auth.models import AbstractUser
from django.db import models

class CustomUser(AbstractUser):
    ROLE_CHOICES = (
        ('student', 'Student'),
        ('teacher', 'Teacher'),
        ('admin', 'Admin'),
    )
    role = models.CharField(max_length=20, choices=ROLE_CHOICES, default='student')

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

Рисунок 2 - Модель данных (Model)

```
from rest_framework.views import APIView
from rest_framework.response import Response
from rest_framework import status
from .serializers import RegisterSerializer

class RegisterView(APIView):
    def post(self, request):
        serializer = RegisterSerializer(data=request.data)
        if serializer.is_valid():
            serializer.save()
            return Response({'message': 'User registered successfully'}, status=status.HTTP_201_CREATED)
        return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)
```

Рисунок 3 - Представления (Views)

Для работы со словарем был разработан модуль, который позволяет преподавателям добавлять слова и комментарии, а студентам — просматривать их. Преподаватели получили доступ к функции добавления новых слов через интерфейс API, а студенты — возможность просматривать готовый словарь (Рисунок 4).

```

# Create your views here.
from rest_framework.views import APIView
from rest_framework.response import Response
from rest_framework import status
from .models import Word, Comment
from .serializers import WordSerializer, CommentSerializer

class WordListCreateView(APIView):
    def get(self, request):
        words = Word.objects.all()
        serializer = WordSerializer(words, many=True)
        return Response(serializer.data)

    def post(self, request):
        serializer = WordSerializer(data=request.data)
        if serializer.is_valid():
            serializer.save()
            return Response(serializer.data, status=status.HTTP_201_CREATED)
        return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)

class CommentCreateView(APIView):
    def post(self, request, word_id):
        try:
            word = Word.objects.get(id=word_id)
        except Word.DoesNotExist:
            return Response({"error": "Word not found"}, status=status.HTTP_404_NOT_FOUND)

        serializer = CommentSerializer(data=request.data)
        if serializer.is_valid():
            serializer.save(word=word)
            return Response(serializer.data, status=status.HTTP_201_CREATED)
        return Response(serializer.errors, status=status.HTTP_400_BAD_REQUEST)

```

Рисунок 4 - Реализация словаря

Было разработано временное решение для обработки текста, которое будет использоваться до полного внедрения алгоритма упрощения. В рамках этой задачи было создано API, которое принимает текст через POST-запрос, передает его в модуль NLP, разработанный Сармусуковой А. С., и возвращает обратно обработанный текст. Это решение позволило протестировать взаимодействие между клиентской и серверной частями платформы, а также интеграцию с предварительным NLP-модулем.

Этап интеграции

На завершающем этапе все компоненты, разработанные членами команды, были интегрированы в общий прототип.

2. 2 Анализ работы

Работая над курсовым проектом, я старалась быть максимально организованной и пунктуальной. Мне удалось достичь всех поставленных целей, хотя изначально задачи казались сложными. В процессе выполнения я столкнулась с рядом трудностей, но поддержка руководителя и интернет ресурсы помогли успешно справиться с проектом.

Я горжусь тем, что смогла разработать модули для регистрации и авторизации пользователей, словаря и API для работы с текстами. Несмотря на ограниченный опыт в реализации подобных проектов, я сумела правильно выбрать технологии и применить их в работе. Особенно приятно осознавать, что созданная мною система работает четко.

Одной из самых больших сложностей стало первоначальное понимание объема задач. Когда я начала работу, я поняла, что многие из них требуют более глубоких знаний, чем те, которыми я обладала. Например, реализация алгоритма упрощения текста была бы слишком сложной и времязатратной. Чтобы преодолеть эту проблему, мы вместе с руководителем решили упростить задачу, создав временное API, возвращающее текст без обработки.

Также сложность вызвала настройка токенов JWT для авторизации. Сначала мне было трудно понять, как правильно их интегрировать в проект. Но, изучив документацию и используя необходимую информацию из Интернета, я смогла успешно внедрить этот метод аутентификации.

Процесс работы был проделан согласно плану с разделением задач. Это помогло мне не потеряться в процессе и сдать все дедлайны в срок.

Работа шла планомерно, и все задачи выполнялись согласно плану. Было уделено время изучению новых технологий и уточнению деталей, что только способствовало улучшению результата.

За время работы над проектом был приобретен ценный опыт в разработке веб-приложений. Я научилась:

- Грамотно организовывать работу с ролями пользователей;
- Использовать JWT для обеспечения безопасной авторизации;
- Работать с API на уровне Django REST Framework;

2.3 Взаимодействие с командой

Работа в команде происходила в формате переписки в общей группе в Telegram. Мы обсуждали текущий статус выполнения задач, делились проблемами. В основном я выполняла все задачи самостоятельно, поэтому взаимодействия с командой было не много.

Руководитель проекта был вовлечён в процесс на всех этапах. Он предоставил материалы для изучения, помог отформатировать задачи под наш уровень знаний и давал обратную связь по промежуточным результатам на протяжении всей работы. Поддержка руководителя мне очень помогала, я чувствовала себя более уверенно.

2.4 Оценка работы руководителя

Я считаю, что руководитель заслуживает высокой оценки. Он грамотно организовал работу команды, своевременно реагировал на возникающие проблемы и информировал нас о приближающихся дедлайнах. Проект был завершён в срок, а все участники команды получили ценный опыт.

3 ЗАКЛЮЧЕНИЕ

Работа над проектом "EduAdapt" позволила создать прототип платформы, предназначенной для адаптации русскоязычных учебных материалов для иностранных студентов. Основная цель проекта состояла в разработке веб-приложения, включающего функции упрощения текста, интерактивного словаря и регистрации пользователей. В процессе реализации были значительные успехи, однако часть функционала остается на стадии доработки.

Цель проекта была достигнута и все задачи были выполнены, создан рабочий прототип платформы, который демонстрирует заявленные функции и может быть использован в образовательных процессах. Основные модули платформы, такие как регистрация и авторизация, словарь и API, были успешно разработаны и интегрированы. Визуальная часть проекта также реализована: макеты интерфейсов разработаны, протестированы и воплощены в программный код. Временное решение для обработки текста, хоть и не является финальным, но обеспечивает базовую функциональность, необходимую для тестирования и демонстрации. Таким образом, проект на текущем этапе может рассматриваться как успешно выполненный.

Все задачи, поставленные передо мной, были выполнены. Был реализован модуль регистрации и авторизации, интерактивный словарь и API для обработки текста. Мой вклад в достижение цели проекта очень велик, потому что моя работа над проектом включала реализацию трех ключевых модулей, без которых невозможно было бы обеспечить функционирование платформы.

Работа над этим проектом позволила мне не только углубить уже полученные ранее знания в области веб-разработки, но и получить опыт

работы в команде, направленной на решение задач социальной значимости, изучить новые возможности языка программирования и применить их на практике. Платформа "EduAdapt" имеет высокий потенциал для дальнейшего развития и может стать полезным инструментом для иностранных студентов, изучающих русскоязычные учебные материалы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Филологи рассказали о сложностях при обучении иностранцев русскому языку // РИА Новости: сайт. – URL: <https://ria.ru/20200606/1572556031.html> (дата обращения: 08.01.2024).

2 Руководство по REST API-дизайну // RESTful Web APIs by Richardson L., Amundsen M., Ruby S: сайт. – URL: <https://www.oreilly.com/library/view/restful-web-apis/9781449359713/> (дата обращения: 20.11.2024)

3 Документация Django // Django Rest Framework: сайт. – URL: <https://www.django-rest-framework.org> (дата обращения: 20.11.2024)

4 Примеры реализации пользовательской аутентификации // Nielsen, J. Usability Engineering. Morgan Kaufmann: сайт. – URL: <https://docs.djangoproject.com> (дата обращения: 23.11.2024)

5 Базовый курс по Django: сайт. – URL: <https://youtube.com/playlist?list=PLF-NY6ldwAWqP4S95brtPHZ5fTCxilgei&si=j5heYiSipYOrlu3n> (дата обращения: 15.11.2024)

6 Углубленный курс по Django: сайт. – URL: <https://youtube.com/playlist?list=PLlWXhlUMyooaDkd39pknA1-Olj54HtpjX&si=hqC7IICQfZYDneSX> (дата обращения: 22.11.2024)

7 Соединении бэкэнда и фронтэнда: сайт. – URL: <https://django.fun/qa/424092/> (дата обращения: 12.12.2024)

8 Паттерн MVC: сайт. – URL: https://skillbox.ru/media/code/chto_takoe_mvc_bazovye_kontseptsii_i_primer_prilozheniya/ (дата обращения: 20.11.2024)

ПРИЛОЖЕНИЕ. ТЕХНИЧЕСКОЕ ЗАДАНИЕ К ПРОЕКТУ

- **Название проекта**

Прототип платформы для адаптации текстов русскоязычных учебных материалов для иностранных студентов «EduAdapt»

- **Цель (назначение)**

Создать прототип платформы для упрощения образовательных материалов для иностранных студентов, обеспечивая удобный доступ к адаптированным текстам и интерактивным инструментам.

- **Сроки выполнения**

Начало 15 ноября 2024 г.

Окончание 20 декабря 2024 г.

- **Руководитель проекта**

Урнежус Филиппс, студент K4240, факультет ИКТ, аналитик Центра карьеры обучающихся ИТМО.

- **Термины и сокращения**

API — Интерфейс программирования приложений, используется для связи между серверной частью (бэкенд) и интерфейсом (фронтенд).

MVC — Архитектурный паттерн «Модель-Представление-Контроллер», помогает разделить логику приложения на три части для упрощения разработки и поддержки.

NLP — Обработка естественного языка, набор технологий для работы с текстами, включая упрощение и анализ сложности текста.

JWT — Токен доступа, используемый для авторизации и поддержания сессии пользователей.

SQLite — Легковесная база данных, используемая для хранения данных в приложении.

- **Технические требования**

Технические:

- Серверная часть построена на Django с использованием Python.
- База данных SQLite для хранения учебных материалов и пользовательских данных.
- REST API для взаимодействия клиентской и серверной частей с использованием Django REST Framework.
- Поддержка кроссбраузерности.

Программные:

- Реализация регистрации, авторизации и функционала с использованием JWT.
- Безопасная валидация данных на стороне сервера для предотвращения SQL-инъекций и XSS-атак.
- Разработка адаптивного интерфейса на Vue.js, совместимого с различными устройствами.

Эргономические:

- Удобное размещение интерфейсных элементов для обеспечения интуитивного взаимодействия пользователей.
- Отдельные интерфейсы для преподавателей и студентов с четким разделением функционала.

- **Содержание работы**

№	Этапы проекта	Сроки выполнения этапов	Ответственный за этап	Вид представления результатов этапа
1	Планирование проекта	15.11.2024 - 18.11.2024	Урнежус Ф.	Готовое техническое задание для дальнейшей работы
2	Разработка серверной части	21.11.2024 - 06.12.2024	Бондар И.А.	Готовый исходный код серверной части прототипа, согласно задачам
3	Разработка интерфейса	25.11.2024 - 12.12.2024	Просветова В. Д.	Разработанные интерфейсы (Регистрация и авторизация; Главная страница с навигацией к разделам сайта; страница с функциями упрощения текста, модуль

				словаря (студент и преподаватель)); соединение интерфейсов с серверной частью проекта.
4	Разработка алгоритма упрощения текста	18.11.2024 - 10.12.2024	Сармусокова А. С.	Реализованный метод упрощения текста с помощью NLP алгоритмов. согласно задаче
5	Тестирование и отладка системы	25.11.2024 - 15.12.2024	Цзян Ю.	Проведенное тестирование на основе составленному списку кейсов по прототипу платформы.
6	Проведение UX-исследования и создание визуальных прототипов	18.11.2024 - 02.12.2024	Грачева С. А.	Анализ и отчет об UX-исследовании на основе трех смежных решений;

				создание макетов страниц в Figma (Регистрация и авторизация; Главная страница с навигацией к разделам сайта; страница с функциями упрощения текста, модуль словаря (студент и преподаватель))
7	Подготовка документации и защита проекта	15.12.2024 - 18.12.2024	Бондар И. А.	Сдача отчета, предоставление доклада с презентацией и демонстрация работы прототипа платформы.

7* Сделать отдельный пункт и расписать задачи для каждого члена команды.

Бондар И. А.

- **Создать API для регистрации и авторизации**

- Реализовать модуль регистрации (логин, пароль, роль) и авторизации (логин и пароль) используя ЯП Python и Django.

- **Разработать API для работы со словарем**

- Реализовать модуль словаря (преподаватель добавляет слово и комментарий, студент просматривает слова) с помощью ЯП Python и библиотекой Django.

- **Разработать API для упрощения текста**

- Необходимо разработать API на языке Python, которое будет выполнять следующую функциональность:
- Принимать текстовые данные: API должно принимать текстовые данные в формате JSON через HTTP-запрос (например, POST-запрос).
- Обращивать текст: Полученный текст должен быть обработан с использованием встроенного NLP-алгоритма. Алгоритм может выполнять такие действия, как анализ, модификация текста, извлечение сущностей, токенизация, лемматизация.
- Возвращать результат: API возвращает результат обработки текста в формате JSON. Результат должен содержать как исходный текст, так и обработанный (если применимо), либо дополнительные данные, полученные в результате обработки.

Просветова В. Д.

- **Реализовать интерфейс для сайта**

- На основе созданных макетов разработать интерфейсы (Регистрация и авторизация; Главная страница с навигацией к

разделам сайта; страница с функциями упрощения текста, модуль словаря (студент и преподаватель)).

- **Интегрировать интерфейс с API**
 - Соединить интерфейсы с API серверной части (CORS).

Сармусокова А. С.

- **Исследовать и разработать функции упрощения текста**
 - Посмотреть и проанализировать готовые решения (библиотеки Python) для упрощения текста и выбрать подходящий для дальнейшей работы. Реализовать метод упрощения в Google Colab.

Грачева С. А.

- **Провести UX-исследование для улучшения удобства интерфейса**
 - Проанализировать и сделать отчет об UX-исследовании на основе трех смежных решений.
- **Создать макеты страниц**
 - Создать макеты страниц в Figma (Регистрация и авторизация; Главная страница с навигацией к разделам сайта; страница с функциями упрощения текста, модуль словаря (студент и преподаватель)).

Цзян Ю.

- **Протестировать API для регистрации и авторизации**
 - Составить список кейсов и провести ручное тестирование по разработанному модулю.
- **Протестировать API для работы со словарем**

- Составить список кейсов и провести ручное тестирование по разработанному модулю.
- **Протестировать API для упрощения текста**
 - Составить список кейсов и провести ручное тестирование по разработанному модулю.
- **Провести финальное тестирование и отладку**
 - Составить список кейсов и провести ручное тестирование по разработанному прототипу платформы.
- **Основные результаты работы и формы их представления**

Готовый прототип платформы EduAdapt представляет собой веб-приложение для адаптации и изучения русскоязычных учебных материалов. Функционал платформы включает модули для упрощения текстов и работы со словарем, обеспечивая удобство и эффективность образовательного процесса.

Функция упрощения текста использует алгоритмы обработки естественного языка (NLP), позволяя автоматически создавать адаптированные версии сложных учебных материалов. Это упрощает понимание текста за счёт использования простого языка, пояснений и дополнительных примечаний.

Словарь представляет собой инструмент, который позволяет преподавателям добавлять новые слова и комментарии, а студентам — просматривать эти записи в адаптированном формате. Он включает интегрированные подсказки, помогающие пользователю понять значение новых слов в контексте учебного материала.

