

**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 **09.03.03 Прикладная информатика**

Образовательная программа **Мобильные и сетевые технологии**

## **КУРСОВОЙ ПРОЕКТ**

Тема: «Разработка модуля обратной связи для обучающей платформы»

Обучающийся: Нгуен Динь Нам K3140

Санкт-Петербург 2024

## СОДЕРЖАНИЕ

<b>ВВЕДЕНИЕ.....</b>	<b>3</b>
1 Актуальность рассматриваемой темы.....	3
2 Цель проекта.....	4
3 Задачи проекта.....	4
<b>ОСНОВНАЯ ЧАСТЬ.....</b>	<b>6</b>
1 Суть проекта и процессы работы над ним.....	6
2 Описать процессы работы над всем проектом.....	7
3 Проблемы и вызовы в процессе разработки.....	10
4 Подходы и процесс выполнения задач.....	11
5 Анализ выполненной работы.....	13
6 Взаимодействие с командой.....	14
7 Взаимодействие с руководителем проекта.....	15
8 Оценка работы руководителя.....	16
<b>ЗАКЛЮЧЕНИЕ.....</b>	<b>18</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....</b>	<b>19</b>
<b>ПРИЛОЖЕНИЕ.....</b>	<b>20</b>

## **ВВЕДЕНИЕ**

### **1 Актуальность рассматриваемой темы**

Современные образовательные технологии развиваются быстрыми темпами, открывая новые возможности для эффективного обучения. В условиях цифровизации образования одной из ключевых задач становится создание платформ, которые не только предоставляют доступ к учебным материалам, но и обеспечивают полноценное взаимодействие между преподавателями и студентами.

Одним из важных компонентов успешного образовательного процесса является организация эффективной и оперативной обратной связи. Отсутствие своевременной обработки и получения обратной связи может привести к целому ряду негативных последствий. Во-первых, это снижает вовлеченность студентов в учебный процесс, так как они не получают необходимую информацию по своей работе и не могут вовремя корректировать ошибки. Во-вторых, преподаватели лишены данных о восприятии учебного материала студентами и о том, какие части курса требуют доработки. Это затрудняет корректировку учебного процесса, что в свою очередь снижает качество образования в целом.

Однако обратная связь является не только важным инструментом для преподавателей, но и полезным для студентов. Она помогает преподавателям отслеживать прогресс студентов, оперативно вносить изменения в курс и улучшать усвоение учебного материала. В то же время, для студентов обратная связь служит источником оценок их работы, рекомендаций по улучшению знаний и навыков, а также разъяснений, которые могут облегчить дальнейшее освоение материала.

Таким образом, создание эффективного модуля обратной связи для образовательных платформ становится актуальной задачей, которая

востребована в учебных заведениях, активно использующих цифровые технологии, а также среди разработчиков образовательного программного обеспечения. Разработка такого модуля способствует улучшению качества образования, повышению удовлетворенности пользователей и улучшению взаимодействия между всеми участниками образовательного процесса.

## **2 Цель проекта**

Целью данного проекта является разработка модуля обратной связи, интегрируемого в существующую цифровую обучающую платформу. Этот модуль должен обеспечивать сбор, обработку и анализ данных, поступающих от пользователей, с целью улучшения качества образовательного процесса. Модуль будет направлен на улучшение взаимодействия между студентами и преподавателями, а также на повышение эффективности и удобства образовательного процесса в целом.

Основные требования к модулю включают:

- Возможность сбора различных типов обратной связи (например, текстовые отзывы, анкеты, оценки);
- Обработка полученной информации для выявления проблемных областей в обучении и корректировки курса;
- Создание удобного интерфейса для пользователей, чтобы они могли легко предоставить и получить обратную связь;
- Интеграция с другими компонентами образовательной платформы для обеспечения ее бесперебойной работы.

## **3 Задачи проекта**

Для достижения поставленной цели необходимо решить несколько ключевых задач, которые охватывают как технические, так и организационные аспекты:

- 1) Анализ требований пользователей.
- 2) Разработка архитектуры модуля.

- 3) Создание пользовательского интерфейса.
- 4) Разработка серверной части.
- 5) Тестирование модуля.
- 6) Анализ и доработка.
- 7) Разработка рекомендаций по развитию модуля.

Таким образом, решение этих задач позволит создать функциональный, удобный и эффективный модуль обратной связи, который будет способствовать повышению качества образования, улучшению коммуникации и взаимодействия между участниками образовательного процесса.

## **ОСНОВНАЯ ЧАСТЬ**

### **1 Суть проекта и процессы работы над ним**

Проект направлен на создание модуля обратной связи, который станет важной частью образовательной платформы. Основная задача модуля заключается в обеспечении эффективного взаимодействия между студентами, преподавателями и администраторами платформы. Этот инструмент позволяет студентам задавать вопросы по учебным материалам, лабораторным работам или работе платформы в целом, а преподавателям и администраторам — получать актуальную информацию о возникающих сложностях и оперативно их решать.

Модуль обратной связи должен предоставлять возможность отправки сообщений с прикрепленными файлами, такими как изображения и документы. Особое внимание уделяется валидации данных: проверяются формат и размер файлов, чтобы исключить возможные ошибки или сбои. Все данные, включая текстовые сообщения и вложения, обрабатываются на стороне сервера и надежно хранятся в базе данных, обеспечивая защиту и структурированность.

Пользовательский интерфейс разработан таким образом, чтобы сделать взаимодействие максимально простым и удобным. Студенты могут быстро заполнить форму, прикрепить необходимые файлы и отправить её на сервер, не сталкиваясь с избыточной сложностью. Одновременно с этим серверная часть обеспечивает эффективную обработку запросов и предоставляет API, что упрощает взаимодействие между фронтендом и бэкендом.

Проект ориентирован на интеграцию с существующей образовательной платформой, а его архитектура позволяет масштабировать решение, если появятся новые требования или увеличится количество пользователей. Кроме того, все компоненты системы упакованы в Docker-контейнеры, что упрощает развертывание, тестирование и поддержку.

Создание такого модуля имеет важное значение для современной образовательной среды. Он не только улучшает взаимодействие между всеми участниками процесса, но и способствует повышению удовлетворенности пользователей платформы. Для команды разработки проект стал не только профессиональным вызовом, но и отличной возможностью применить современные технологии, такие как Fast API [5], PostgreSQL и Docker, в реальной задаче.

## **2 Описать процессы работы над всем проектом**

Процесс разработки модуля обратной связи для обучающей платформы был организован в несколько ключевых этапов, каждый из которых играл важную роль в достижении конечной цели.

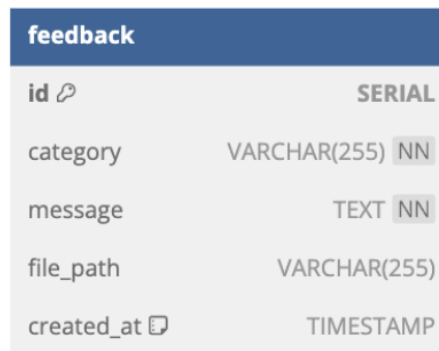
Сначала мы начали с тщательного исследования и анализа существующих решений на рынке. Мы обратили внимание на интерфейсы, удобство использования и отзывы пользователей. Это позволило нам выявить лучшие практики и недостатки текущих систем, а также понять, какие функции наиболее востребованы среди пользователей.

Мы также провели опросы среди студентов и преподавателей, чтобы собрать информацию о их потребностях и ожиданиях. Вопросы касались того, какие функции они хотели бы видеть в модуле обратной связи, как они предпочитают взаимодействовать с преподавателями и какие проблемы они сталкиваются в текущих системах. Результаты опросов помогли нам сформировать четкое представление о том, что именно необходимо для успешной реализации проекта.

После сбора информации мы перешли к проектированию архитектуры модуля. На этом этапе была разработана ER-диаграмма базы данных, которая описывает структуру хранения данных, включая таблицы для сообщений, пользователей и статусов. Мы определили, какие данные будут храниться в базе данных и как они будут связаны друг с другом. Это обеспечивало четкое понимание того, как данные будут организованы и как будет происходить

взаимодействие между фронтендом и бэкендом. Результат работы изображен на рисунке 1.

## ER-диаграмма БД



feedback	
id	SERIAL
category	VARCHAR(255) NN
message	TEXT NN
file_path	VARCHAR(255)
created_at	TIMESTAMP

Рисунок 1 — ER-диаграмма таблицы feedback

Кроме того, мы разработали спецификации для API, которые будут использоваться для взаимодействия между клиентской и серверной частями. Это включало определение необходимых эндпоинтов, методов HTTP и форматов данных, которые будут передаваться. Мы также обсудили вопросы безопасности, такие как аутентификация пользователей и защита данных.

В процессе разработки бэкенда мы выбрали Fast API как основной фреймворк для создания API. Это решение было основано на его высокой производительности и простоте использования. Я занимался реализацией основных функций, таких как обработка запросов на отправку сообщений и загрузку файлов. Мы также внедрили валидацию данных, чтобы гарантировать, что все входящие данные соответствуют установленным требованиям. Это включало проверку обязательных полей, длины сообщений и форматов загружаемых файлов.

Параллельно с разработкой бэкенда я использовал React JS для создания интерфейса, основываясь на компонентах, разработанных с использованием Astro [3]. Я разработал формы для отправки сообщений и



загрузки файлов, обеспечивая их корректное отображение на различных устройствах. Использование ReactJS позволило мне создать динамичные и отзывчивые интерфейсы, которые легко адаптируются к различным размерам экранов.

Важным этапом была интеграция фронтенда с бэкэндом, где я использовал объект FormData для отправки данных на сервер. Это позволило нам легко обрабатывать как текстовые сообщения, так и файлы, прикрепленные пользователями. Я также реализовал функции предварительного просмотра файлов, что помогло пользователям проверить содержимое перед отправкой.

На этапе тестирования мы провели как автоматизированные, так и ручные тесты, чтобы убедиться в корректности работы всех функций. Я занимался тестированием загрузки файлов и отправки сообщений, что позволило выявить и исправить ошибки до запуска. Мы также организовали сессии обратной связи с пользователями, чтобы получить их мнение о функциональности и удобстве использования модуля.

Мы провели тестирование производительности, чтобы убедиться, что модуль может обрабатывать большое количество запросов одновременно без сбоев. Это было очень важно, так как мы ожидали, что модуль будет использоваться многими студентами и преподавателями одновременно.

После завершения всех этапов разработки и тестирования мы подготовили модуль к запуску на обучающей платформе. Я также участвовал в организации обучающих сессий для студентов и преподавателей, чтобы помочь им освоить новый функционал и ответить на возникающие вопросы. Мы создали документацию, в которой подробно описывались все функции модуля и инструкции по его использованию.

Таким образом, процесс работы над проектом был организован последовательно и структурировано, что позволило нам успешно реализовать

модуль обратной связи и обеспечить его функциональность в соответствии с требованиями пользователей.

### **3 Проблемы и вызовы в процессе разработки**

В процессе работы над проектом "Разработка модуля обратной связи для обучающей платформы" я столкнулся с несколькими основными проблемами, которые требовали решения. Одна из самых больших проблем заключалась в недостаточной ясности в использовании FormData для отправки данных с пользовательского интерфейса на сервер. Хотя у меня были базовые знания HTML и JavaScript [4], интеграция FormData в процесс отправки сообщений и загрузки файлов стала для меня новой задачей.

Когда я начал разрабатывать пользовательский интерфейс с использованием React JS [2], я понял, что использование FormData необходимо для эффективной обработки вложенных файлов и данных формы. Одной из основных причин использования FormData вместо JSON.stringify является возможность обработки двоичных файлов, таких как изображения или документы. FormData позволяет мне легко отправлять эти файлы, не преобразуя их в строку, что может вызвать проблемы с форматированием и размером.

Тем не менее, я столкнулся с трудностями в понимании того, как работает FormData, как создать объект FormData и как отправить его через API на бэкэнд. Это замедлило процесс разработки и вызвало у меня неуверенность в правильности подхода.

Кроме того, я столкнулся с проблемой управления состоянием форм в ReactJS. Когда пользователи вводят данные в поля формы, мне нужно было убедиться, что состояние этих полей обновляется правильно и синхронизируется с данными, которые вводят пользователи. Это иногда создавало сложности, особенно когда было много полей данных и нужно было обрабатывать события, такие как изменение значений или отправка формы.

Еще одной проблемой, с которой я столкнулся, было привыкание к организации и структуре кода в React [2]. Когда я только начинал, мне было трудно понять, как разбить приложение на компоненты и как управлять props и state между этими компонентами. Это иногда приводило к тому, что код становился трудночитаемым и трудным для поддержки.

Обработка жизненного цикла компонентов также была вызовом. Мне нужно было понять, как использовать хуки, такие как use Effect, для выполнения задач, таких как вызов API или обновление состояния, когда компонент монтируется или размонтируется из DOM. Это требовало от меня глубокого понимания того, как работает React и как взаимодействуют хуки.

Кроме того, я столкнулся с трудностями в обработке ошибок и уведомлении пользователей. При отправке данных, если возникала ошибка на стороне сервера, мне нужно было убедиться, что пользователи получают четкие и понятные уведомления. Это требовало от меня создания эффективной и удобной системы уведомлений.

Наконец, оптимизация производительности приложения также была проблемой, на которую мне нужно было обратить внимание. Работая с множеством компонентов и большими объемами данных, мне нужно было изучить, как оптимизировать рендеринг и минимизировать количество ненужных вызовов API, чтобы улучшить пользовательский опыт.

Эти вызовы создали напряженную рабочую среду, требуя от меня быстрого обучения и адаптации, чтобы гарантировать, что модуль обратной связи может эффективно работать и удовлетворять потребности пользователей.

#### **4 Подходы и процесс выполнения задач**

В рамках проекта "Разработка модуля обратной связи для обучающей платформы" я выполнил несколько ключевых задач, которые позволили успешно реализовать проект.

#### **4.1 Изучение работы с FormData**

В рамках первого этапа я сосредоточился на изучении работы с формами и объектом FormData [1]. Я изучил, как создавать, заполнять и отправлять данные на сервер, а также обрабатывать вложения. Для этого я провел эксперименты с простыми примерами, что позволило мне уверенно использовать FormData в проекте.

#### **4.2 Разработка поля для загрузки файлов**

На следующем этапе я приступил к разработке поля для загрузки файлов в форме. Я создал компонент на React [6], который позволял пользователям загружать файлы [2]. Я применил различные методы для обеспечения корректной работы поля ввода, включая проверку типа и размера загружаемого файла. Когда пользователь выбрал файл, я выводил информацию о нем в консоль браузера, чтобы убедиться, что файл был загружен успешно.

#### **4.3 Отправка файла на бэкенд**

После завершения разработки поля для загрузки файлов я сосредоточился на отправке файла (изображения) на бэкенд. Я использовал FormData для упаковки файла и отправки его на сервер через HTTP-запрос [8]. Важно было убедиться, что сервер правильно обрабатывает полученные данные и возвращает положительный ответ. Я также реализовал обработку ошибок, чтобы пользователи получали уведомления о статусе отправки данных, включая сообщения об успешной отправке или возникших ошибках.

#### **4.4 Интеграция с API и тестирование**

На заключительном этапе я сосредоточился на интеграции с API. Я создал функции для обработки запросов, которые отправляли данные на сервер и обрабатывали ответы. Я убедился, что все данные отправляются корректно и что пользователи получают актуальную информацию о статусе своих запросов. В завершение, я провел тестирование и отладку всех функций, чтобы убедиться, что они работают корректно. Я проверил отправку

сообщений и загрузку файлов, а также обработку ошибок [7]. В процессе тестирования я выявил и исправил несколько ошибок, что позволило улучшить стабильность и производительность модуля. Результат работы представлен на рисунке 2.

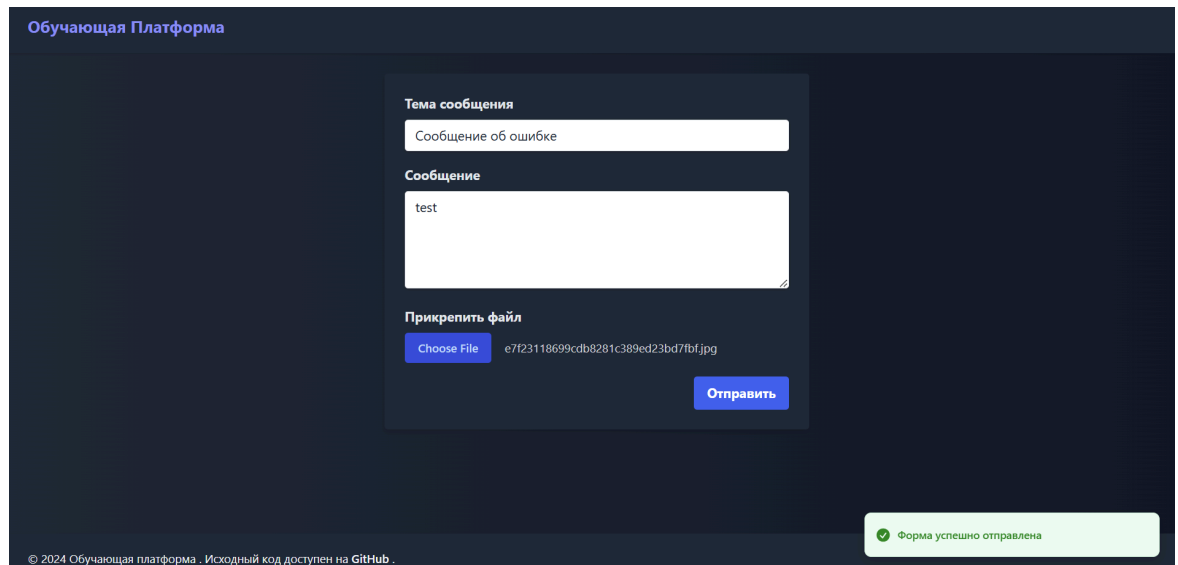


Рисунок 2 — Результат завершения модуля обратной связи

## 5 Анализ выполненной работы

В этом разделе я проанализирую свой опыт работы над проектом "Разработка модуля обратной связи для обучающей платформы", выделяя как достижения, так и трудности, с которыми я столкнулся в процессе реализации.

### 5.1 Успехи в проекте

В процессе реализации проекта я успешно завершил основные функции модуля. Я создал удобный интерфейс, который позволяет пользователям легко загружать файлы и получать обратную связь о статусе своих запросов. Этот интерфейс не только удовлетворяет потребности пользователей, но и делает их взаимодействие с системой более комфортным.

### 5.2 Проблемы и трудности

Тем не менее, не все прошло гладко. Я столкнулся с некоторыми трудностями, особенно в процессе интеграции с API. Обработка ответов от сервера иногда не соответствовала ожиданиям, и мне приходилось тратить

много времени на отладку и выяснение причин этих ошибок. Это вызывало у меня замешательство и влияло на темп работы. Кроме того, работа с FormData и отправка файлов на сервер стали для меня значительным вызовом, так как у меня не было достаточного опыта в этой области.

### **5.3 Планомерная работа**

Хотя я старался следовать плану, некоторые факторы мешали этому. Непредвиденные ошибки и проблемы с интеграцией иногда отвлекали меня от первоначального плана. Я также заметил, что нехватка времени на тестирование и отладку приводила к спешке, что могло негативно сказаться на качестве конечного продукта.

### **5.4 Извлеченные уроки**

Тем не менее, в процессе выполнения проекта я усвоил множество полезных уроков. Я улучшил свои навыки работы с React и FormData, а также научился более эффективно интегрировать фронтенд и бэкенд. Я также осознал важность тестирования и отладки для обеспечения стабильной работы продукта и хорошего пользовательского опыта. Этот проект не только помог мне развить навыки, но и предоставил ценный опыт, который я смогу использовать в будущих проектах.

## **6 Взаимодействие с командой**

В процессе реализации проекта "Разработка модуля обратной связи для обучающей платформы" я имел возможность активно взаимодействовать с моими однокурсниками, которые также являются студентами. Мы организовывали еженедельные встречи, на которых обсуждали трудности, с которыми каждый из нас сталкивался в своих задачах, и предлагали возможные решения.

Каждая встреча проходила в открытой и дружелюбной атмосфере, где каждый мог поделиться своими мнениями и предложениями. Мы совместно анализировали проблемы и искали эффективные решения. Это не только

помогало нам справляться с трудностями, но и способствовало укреплению командного духа.

Кроме того, на встречах мы регулярно обновляли информацию о ходе выполнения задач каждого участника. Это позволяло всем быть в курсе общей ситуации по проекту и оказывать поддержку друг другу при необходимости. Я чувствовал, что сотрудничество и поддержка со стороны моих товарищей по команде помогли мне улучшить навыки командной работы и управления временем.

Я также многому научился у своих однокурсников, начиная от подходов к решению проблем и заканчивая различными техниками программирования. Это взаимодействие не только повысило качество нашего проекта, но и способствовало моему личному и профессиональному развитию.

## **7 Взаимодействие с руководителем проекта**

В ходе реализации моего проекта по разработке модуля обратной связи для обучающей платформы я имел возможность активно взаимодействовать с моим руководителем, Жуковым Вадимом Витальевичем. Я часто задавал ему вопросы о материалах по ReactJS, о том, как отправлять данные с помощью FormData, а также о других трудностях, с которыми сталкивался в процессе разработки.

Жуков всегда быстро отвечал и оказывал мне теплую поддержку. Его готовность помочь создала позитивную учебную атмосферу, в которой я чувствовал себя комфортно, задавая вопросы и ища помощь. Он не только отвечал на мои вопросы, но и указывал на ошибки, которые я допускал, предлагая эффективные способы их исправления. Это помогло мне не только исправить недочеты, но и лучше понять процесс разработки и оптимизации кода.

Кроме того, он регулярно напоминал мне и проверял мой прогресс. Когда я сталкивался с трудностями, он не только помогал мне решить

проблемы, но и предоставлял полезные материалы для самостоятельного изучения и улучшения моих навыков.

Это взаимодействие не только помогло мне преодолеть трудности в проекте, но и способствовало моему личному и профессиональному развитию. Я очень ценю поддержку и наставничество Жукова, и уверен, что знания и опыт, которые я получил от него, станут прочной основой для будущих проектов.

## **8 Оценка работы руководителя**

Работа нашего руководителя, Жукова Вадима Витальевича, была исключительно полезной и эффективной на протяжении всего проекта. Его профессионализм и опыт в области разработки программного обеспечения значительно способствовали обучению и развитию всей нашей группы.

Во-первых, Жуков всегда был доступен для консультаций и ответов на наши вопросы. Его быстрая реакция и готовность помочь создали атмосферу доверия и поддержки, что позволило нам чувствовать себя уверенно в процессе работы. Он не только предоставлял необходимые материалы и ресурсы, но и делился своими знаниями, что помогало нам лучше понять сложные аспекты проекта.

Во-вторых, его способность давать конструктивную обратную связь была очень ценна. Он указывал на ошибки, которые допускала наша группа, и предлагал конкретные способы их исправления, что способствовало профессиональному росту каждого члена команды. Благодаря его рекомендациям, мы смогли улучшить свои навыки программирования и научиться более эффективно решать возникающие проблемы.

Кроме того, он регулярно следил за нашим прогрессом и проверял выполнение задач. Это помогало нам оставаться на правильном пути и не упускать важные детали. Его внимание к деталям и стремление к качеству работы вдохновляли нас на достижение высоких результатов.



В целом, мы считаем, что работа Жукова Вадима Витальевича была неоценимой для успеха нашей группы в проекте. Его поддержка, знания и опыт значительно способствовали нашему обучению и развитию как команды разработчиков. Мы очень благодарны ему за все, что он сделал для нас, и уверены, что его наставничество окажет положительное влияние на карьеру каждого члена группы в будущем.

## **ЗАКЛЮЧЕНИЕ**

Проект достиг поставленных целей и успешно выполнил все задачи. Разработанный нашей группой модуль обратной связи был спроектирован и реализован в соответствии с требованиями пользователей, а также прошел тщательное тестирование, чтобы гарантировать его функциональность и удобство. Конечный продукт не только соответствует ожиданиям пользователей, но и обеспечивает положительный опыт взаимодействия с ним.

Тем не менее, в процессе работы мы столкнулись с определенными трудностями. Одной из самых больших проблем было то, что некоторые члены команды испытывали сложности в освоении новых технологий, особенно ReactJS. Это потребовало от нас дополнительного времени для обучения и адаптации к этим технологиям. Однако благодаря поддержке нашего руководителя, Жукова Вадима Витальевича, и неустанным усилиям всей группы, мы смогли преодолеть эти препятствия и завершить проект в срок.

Мой вклад в успех проекта заключался в активном участии как в процессе разработки, так и в тестировании модуля. Я взял на себя реализацию функций, связанных с отправкой данных через FormData, а также работал над улучшением пользовательского интерфейса, чтобы он был максимально удобным и интуитивно понятным. Кроме того, я активно участвовал в групповых обсуждениях, высказывая свои идеи и предлагая решения для возникающих проблем, что также способствовало общему успеху команды.

Полученные знания и опыт станут надежной основой для меня в будущих проектах. Я уверен, что то, что мы узнали и пережили, поможет нам достичь новых высот в нашей карьере в области разработки программного обеспечения.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Использование объектов FormData - [https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest\\_API/Using\\_FormData\\_Objects#creating\\_a\\_formdata\\_object\\_from\\_scratch](https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest_API/Using_FormData_Objects#creating_a_formdata_object_from_scratch)
2. Хук useReducer в React - <https://react.dev/reference/react/useReducer>
3. Гидратация интерактивных компонентов в Astro - <https://docs.astro.build/en/guides/framework-components/#hydrating-interactive-components>
4. Работа с формами в JavaScript - <https://doka.guide/js/deal-with-forms/>
5. Первый шаги с FastAPI - <https://fastapi.tiangolo.com/ru/tutorial/first-steps/>
6. Документация по React - <https://legacy.reactjs.org/docs/getting-started.html>
7. Обработка ошибок в JavaScript - [https://developer.mozilla.org/en-US/docs/Learn\\_web\\_development/Extensions/Async\\_JS/Promises#error\\_handling](https://developer.mozilla.org/en-US/docs/Learn_web_development/Extensions/Async_JS/Promises#error_handling)
8. Работа с API в JavaScript - [https://developer.mozilla.org/en-US/docs/Web/API/Fetch\\_API/Using\\_Fetch](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch)

## ПРИЛОЖЕНИЕ

### ТЕХНИЧЕСКОЕ ЗАДАНИЕ

#### 1 Название проекта

Модуль обратной связи для обучающей платформы

#### 2 Цель (назначение)

Целью данного проекта является разработка модуля обратной связи для обучающей платформы, который позволит пользователям оставлять отзывы о курсах и задавать вопросы. Этот модуль будет служить эффективным инструментом для взаимодействия между пользователями и администрацией, способствуя повышению уровня вовлеченности и удовлетворенности.

Модуль также будет способствовать формированию сообщества, где пользователи могут обмениваться мнениями и опытом, создавая более поддерживающую образовательную среду.

#### 3 Сроки выполнения

Начало: 01 ноября 2024 г.

Окончание: 20 декабря 2024 г.

#### 4 Руководитель проекта

Жуков Вадим Витальевич

#### 5 Термины и сокращения

**API - Application Programming Interface (интерфейс прикладного программирования).** Это набор правил и протоколов, позволяющий различным приложениям взаимодействовать друг с другом. Он упрощает обмен данными и функциональностью, позволяя разработчикам интегрировать сторонние сервисы без необходимости знать их внутреннюю структуру.

**JSON - JavaScript Object Notation.** Легкий формат обмена данными, удобный для чтения и записи человеком, а также для парсинга и генерации машинами.

**Swagger/OpenAPI.** Инструменты для документирования и описания API, позволяющие разработчикам создавать, тестировать и документировать API.

**СУБД - Система управления базами данных.** Программное обеспечение, которое позволяет создавать, управлять и манипулировать базами данных.

**QA - Quality Assurance (обеспечение качества).** Процесс, направленный на улучшение и поддержание качества продукта на всех этапах разработки.

**YAML - Yet Another Markup Language (формат разметки).** Читаемый человеком формат сериализации данных, часто используемый для конфигурационных файлов.

**Бэклог.** Очередь задач, которые необходимо выполнить в рамках проекта, обычно включает в себя функции, исправления ошибок и улучшения.

**Деплой.** Процесс переноса кода из среды разработки на рабочий сервер, чтобы сделать его доступным для пользователей.

**Прод/продакшн.** Завершающий этап разработки, когда программа проходит сборку, тестирование и развертывание на рабочем сервере.

**Технологический стек.** Набор инструментов и технологий, необходимых для разработки продукта, включая языки программирования, фреймворки и библиотеки.

**Фреймворк.** Готовая структура и набор инструментов, на основе которых ведется разработка на разных языках программирования, упрощая процесс создания приложений.

## **6 Технические требования**

### **6.1 Требования к системе в целом**

Производительность: Модуль должен обрабатывать не менее 100 запросов в минуту с временем отклика не более 2 секунд.

Надежность: Доступность системы должна составлять не менее 99,5% в месяц.

Совместимость: Система должна быть совместима с последними версиями браузеров (Chrome, Firefox, Safari, Edge) и мобильными устройствами.

### **6.2 Требования к структуре и функционированию системы**

Архитектура: Система должна быть построена на основе микро сервисной архитектуры для обеспечения гибкости и масштабируемости.

Подсистемы: Система должна включать следующие подсистемы:

- Frontend: Интерфейс пользователя для ввода данных.
- Backend: Обработка и хранение данных.

### **6.3 Требования к функциям (задачам), выполняемым системой**

Таблица 1 - Требования к техническим функциям системы

Наименование	База данных	Заказчики
Обработка и сохранение данных формы (Backend)	Локально/PostgreSQL	Разработчики
Swagger/OpenAPI документация для API (Backend)	Нет	Разработчики
Эндпоинт для отправки данных формы (Backend)	Нет	Разработчики

Поля ввода контактных данных и сообщения (Frontend)	Нет	Все пользователи
Валидация данных формы перед отправкой на сервер (Front End)	Нет	Все пользователи
Контейнеризация приложения с помощью Docker (DevOps)	Нет	Разработчики
Деплой приложения в облако (DevOps)	Нет	Разработчики
Чек-лист и тест-кейсы на разработанный функционал (QA)	Нет	Разработчики
Руководство пользователя по использованию приложения (Общее/QA)	Нет	Все пользователи

#### **6.4 Требования к безопасности**

Защита данных: Все данные пользователей должны быть зашифрованы при передаче (использование HTTPS) и хранении.

Аутентификация: Система должна поддерживать многофакторную аутентификацию для повышения уровня безопасности.

#### **6.5 Требования к эргономике и технической эстетике**

Интерфейс пользователя: Дизайн интерфейса должен быть интуитивно понятным и удобным для пользователей, с учетом принципов UI/UX.

Доступность: Система должна соответствовать стандартам доступности (WCAG 2.1).

#### **6.6 Дополнительные требования**

Документация: Необходимо предоставить полную документацию по использованию системы, включая руководство пользователя и техническую документацию.

Тестирование: Разработать чек-лист и тест-кейсы для проверки всех ключевых функций системы.

## 7 Содержание работы

Таблица 2 - Этапы проекта и сроки их выполнения

№	Этапы проекта	Сроки выполнения этапов	Ответственный за этап	Вид представления результатов этапа
1	Разработка технического задания	1-10 ноября	Жуков Вадим Витальевич	
2	Настроить генерацию документации API (Swagger/OpenAPI)	11-24 ноября	Сафонова Людмила Марковна	Презентация
3	Изучить работу с формами (FormData)	11-17 ноября	Нгуен Динь Нам	Презентация
4	Изучить Fast API и развернуть стартовое приложение	11-17 ноября	Плешнев Арсений Вадимович	Презентация
5	Создать новый проект Astro	11-17 ноября	Колесников Игорь Евгеньевич	Презентация
6	Создать эндпоинты для отправки и обработки данных формы (Fast API)	18-24 ноября	Плешнев Арсений Вадимович	Презентация
7	Настроить сохранение данных в базе данных или локально	25 ноября - 8 декабря	Плешнев Арсений Вадимович	Презентация
8	Верстка формы обратной связи	18-24 ноября	Колесников Игорь Евгеньевич	Презентация



9	Настроить валидацию данных на клиенте	25 ноября - 1 декабря	Колесников Игорь Евгеньевич	Презентация
10	Создать поля для ввода текста и контактных данных	18-24 ноября	Колесников Игорь Евгеньевич	Презентация
11	Настроить отправку данных на бэкенд через API	2-8 декабря	Колесников Игорь Евгеньевич	Презентация
12	Настроить контейнеризацию приложения (бэкенд)	25 ноября - 1 декабря	Сафонова Людмила Марковна	Презентация
13	Настроить контейнеризацию приложения (фронтенд)	9-15 декабря	Колесников Игорь Евгеньевич	Презентация
14	Сверстать поле для загрузки файлов в форме	25-30 ноября	Нгуен Динь Нам	Презентация
15	Отправить файл (картинку) на бэкенд	1-10 декабря	Нгуен Динь Нам	Презентация
16	Протестировать интеграцию фронтенда и бэкенда	25 ноября - 1 декабря	Юров Кирилл Игоревич	Презентация
17	Написать чек-лист и тест-кейсы	11-24 ноября	Юров Кирилл Игоревич	Презентация
18	Перенести код из среды разработки на сервер	2-8 декабря	Сафонова Людмила Марковна	Презентация
19	Подготовить руководство пользователя	2-8 декабря	Юров Кирилл Игоревич	Презентация

20	Защита проекта (сдача отчета и представление доклада с презентацией)	20 декабря	Арсений, Людмила, Игорь, Нам, Кирилл, Максин	Отчёт о проделанной работе, презентация.
----	--	------------	--	--

## 8 Основные результаты работы

Рабочее веб-приложение с формой обратной связи: Разработано функциональное веб-приложение, позволяющее пользователям оставлять отзывы и получать ответы на свои вопросы.

Подготовлена документация: Создана полная документация к разработанному функционалу, включая API и руководство пользователя, что обеспечивает удобство в использовании и интеграции.

Составлены тест-кейсы: Разработаны тест-кейсы для проверки ключевых функций приложения, что гарантирует его надежность и стабильность.

Упаковка в Docker-контейнеры: Приложение упаковано в Docker-контейнеры, что обеспечивает его переносимость и легкость в развертывании.

Настройка управления через Docker Compose: Управление приложением настроено через Docker Compose, что упрощает процесс развертывания и управления зависимостями.

Деплой в облачную среду: Выполнен успешный деплой приложения в облачную среду, что обеспечивает его доступность и масштабируемость для пользователей.