

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования  
«Уральский федеральный университет  
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ  
Школа бакалавриата

## ОТЧЕТ

По проекту  
«Название проекта»

по дисциплине «Проектный практикум»

Заказчик: Смирнов Д.С

Куратор: Пушкарь Ю.

Студенты команды \_\_\_\_\_ Pa&SE\_\_\_\_\_

МАТОВИЧ С.

ШАЙХУТДИНОВ К.Ю.

ХАНОВ Э.Р.

ГУЩИН М.А.

ЗАРИПОВ З.Р.

Екатеринбург, 2025

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1.1 Проблема стейкхолдеров.....	5
1.2 Цель проекта .....	5
1.3 Конкуренты.....	5
1.4 Создание надежного механизма для ввода и хранения информации о вузах и работодателях.....	6
1.4.1 Разработка гибкой возможности загрузки и/или создания шаблонов договоров/приложений к практике: .....	6
1.4.2 Реализация интеллектуальной автоматической подстановки реквизитов практикантов в шаблоны документов: .....	6
1.4.3 Создание интуитивно понятного механизма для ввода студентами данных о своей практике: .....	7
1.4.4 Обеспечение безупречного формирования заполненных печатных форм приложений к договорам: .....	7
1.5 Актуальность и важность проекта.....	7
1.6 Область применения программного продукта.....	8
1.7 Ожидаемые результаты и планируемые достижения по завершении проекта: .....	8
2 Основная часть .....	9
2.1 информацию о работе каждого участника в отдельности; .....	9
2.1.1 Аналитик и тим-лид(Матович Стефан) .....	9
2.1.2 Дизайнер и разработчик Front-end(ХАНОВ ЭМИЛЬ РУСТЕМОВИЧ).....	9
2.1.3 Фронтенд-разработчик(Шайхутдинов Кирилл Юрьевич).....	9
2.1.4 Главный Back-end разработчик(Гущин Максим Александрович)..	10
2.1.5 Второй Back-end разработчик(ЗАРИПОВ ЗАГИР РАИСОВИЧ)	10
2.2 разбор требований заказчика и пользователей к программному продукту .....	10

2.3 составление плана действий для достижения цели (backlog); .....	11
2.4 анализ и сопоставление аналогов разрабатываемого продукта .....	12
2.4.1 Управление руководителями практики: .....	12
2.4.2 Гибкость настройки форм: .....	13
2.5 обзор архитектуры программного продукта, описание основных компонентов и связей между ними, обоснование выбора архитектурного решения .....	13
2.5.1 Сервис аутентификации .....	13
2.5.2 Сервис заявок .....	13
2.5.3 Сервис генерации .....	14
2.5.4 Технологии и архитектура: .....	14
2.5.5 Причины выбора технологий.....	14
2.6 описание методологии разработки и информация о процессе разработки .....	14
2.6.1 описание методологии разработки.....	15
2.6.2 информация о процессе разработки:.....	15
2.7 отчет о результатах тестирования на промежуточных этапах, разбор выявленных ошибок.....	16
2.8 информация о планировании деятельности в ходе разработки и распределении задач между участниками команды разработчиков.....	17
3 ЗАКЛЮЧЕНИЕ .....	18
3.1 Оценка соответствия программного продукта поставленным требованиям заказчика .....	18
3.2 оценка качества программного продукта на основе результатов тестирования .....	19
3.3 Вывод.....	19
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	20
ПРИЛОЖЕНИЕ А .....	22
ПРИЛОЖЕНИЕ В .....	22

## **ВВЕДЕНИЕ**

В современном мире эффективность и автоматизация рабочих процессов играют ключевую роль в успехе любой организации. Данный проект направлен на решение актуальной задачи для НПЦ «РИЦ» — оптимизацию и автоматизацию процесса оформления договоров о практике для студентов и администраторов. Существующая процедура, предполагающая ручное заполнение и обработку большого количества документов, отнимает много времени и ресурсов, а также сопряжена с риском ошибок. Целью проекта является разработка прототипа модуля работы с партнёрами, который позволит НПЦ «РИЦ» перейти от использования устаревших методов, таких как гугл-документы, к современной, централизованной и автоматизированной системе. В частности, модуль позволит организатору практики создавать и загружать в веб-сервис формы приложений к договорам, адаптированные к требованиям различных вузов-партнёров. Студенты, в свою очередь, смогут заполнять эти формы через удобный веб-интерфейс, внося необходимые данные о своей практике: ФИО, тип и сроки практики, руководитель практики. По окончании периода записи на практику администратор сможет сгенерировать печатную форму приложения/договора, готовую к подписанию и отправке в соответствующий вуз. Проект охватывает широкий спектр задач: от разработки пользовательского интерфейса и серверной логики до интеграции с существующими базами данных и создания механизма генерации печатных форм. Успешная реализация прототипа позволит НПЦ «РИЦ» значительно сократить время и ресурсы, затрачиваемые на оформление договоров о практике, повысить точность данных и улучшить взаимодействие со студентами и партнерами.

# **1 Аналитика**

## **1.1 Проблема стейкхолдеров**

Использование Google-документов для оформления практики создает ряд проблем: опасения организатора по поводу целостности данных студентов, риски нарушения конфиденциальности, вероятность ошибок при заполнении студентами и страх потери данных из-за действий других пользователей. Существующая система подвержена риску нарушения целостности, конфиденциальности и доступности данных. А так же проблема с неудобством работы заказчиком с Google-документами.

## **1.2 Цель проекта**

Основная цель данного проекта — разработка веб-ориентированной подсистемы «Партнеры», призванной радикально изменить процесс оформления договоров о прохождении практики в НПЦ «РИЦ». Эта подсистема станет мощным инструментом автоматизации, позволяющим существенно сократить время и трудозатраты, связанные с вводом данных и формированием печатных форм приложений к договорам. Внедрение подсистемы «Партнеры» позволит повысить точность и надежность информации, минимизировать риск ошибок, а также улучшить взаимодействие со студентами и партнерами.

## **1.3 Конкуренты**

Поскольку разрабатываемый продукт является составной частью более крупной системы и не представляет собой независимый проект, прямое сравнение с масштабными аналогами нецелесообразно. Основным конкурентом является другая команда, работающая над аналогичным функционалом.

Наши преимущества:

- 1) это реализация редактирования формы студентов ,организатором практики

- 2) Добавление и редактирование руководителей практики
- 3) Дизайн сделан под формат сайта “Точки сбора ”

## **1.4 Создание надежного механизма для ввода и хранения информации о вузах и работодателях**

Для достижения этой амбициозной цели необходимо решить ряд взаимосвязанных задач, каждая из которых вносит свой вклад в общую функциональность и эффективность подсистемы

эта задача предполагает разработку удобного и интуитивно понятного интерфейса для добавления, редактирования и хранения данных о партнерах НПЦ «РИЦ». Важно обеспечить возможность структурированного хранения информации, включая контактные данные, реквизиты и другие необходимые сведения.

### **1.4.1 Разработка гибкой возможности загрузки и/или создания шаблонов договоров/приложений к практике:**

эта задача требует создания функциональности, позволяющей администраторам загружать в систему уже существующие шаблоны документов, а также создавать новые шаблоны непосредственно в веб-интерфейсе. Важно предусмотреть возможность адаптации шаблонов к требованиям различных вузов и организаций.

### **1.4.2 Реализация интеллектуальной автоматической подстановки реквизитов практикантов в шаблоны документов:**

эта задача предполагает разработку алгоритма, который автоматически заполняет шаблоны договоров и приложений данными студентов, такими как

ФИО, дата рождения, специальность и другие необходимые сведения. Автоматическая подстановка реквизитов позволит избежать ручного ввода данных, значительно сократить время оформления документов и минимизировать риск ошибок.

#### **1.4.3 Создание интуитивно понятного механизма для ввода студентами данных о своей практике:**

эта задача требует разработки пользовательского интерфейса, который позволит студентам легко и быстро вводить необходимую информацию о своей практике, такую как место прохождения практики, сроки практики, руководитель практики и другие сведения.

#### **1.4.4 Обеспечение безупречного формирования заполненных печатных форм приложений к договорам:**

эта задача предполагает разработку механизма, который автоматически формирует готовые к печати приложения к договорам, содержащие всю необходимую информацию о студенте и его практике. .

Решение всех этих задач позволит создать комплексную и эффективную веб-ориентированную подсистему «Партнеры», которая значительно упростит и автоматизирует процесс оформления договоров о прохождении практики в НПЦ «РИЦ», повысит качество данных и улучшит взаимодействие со студентами и партнерами.

### **1.5 Актуальность и важность проекта**

Проект актуален в связи с необходимостью оптимизации текущего процесса оформления договоров о практике. Ручной ввод данных,

необходимость индивидуального оформления документов для каждого студента и использование сторонних инструментов (например, Google Docs) приводят к увеличению трудозатрат, возможности ошибок и задержкам в обработке данных. Автоматизация этих процессов позволит сократить время, необходимое для оформления документов, уменьшить количество ошибок, повысить эффективность работы администраторов и улучшить общее взаимодействие с партнерами и студентами.

### **1.6 Область применения программного продукта**

разрабатываемый модуль предназначен для использования в НПЦ «РИЦ» сотрудниками, ответственными за организацию практики студентов. Он может быть интегрирован с существующими веб-сервисами организации (Точка сбора) или развернут как отдельный сайт. Потенциальными пользователями являются администраторы практики, отвечающих за работу с учебными заведениями, а также студенты, проходящие практику.

### **1.7 Ожидаемые результаты и планируемые достижения по завершении проекта:**

В рамках данного проекта планируется достичь следующих результатов: Разработана подсистема «Партнеры», включающая базу данных для хранения заполненных студентами форм и шаблонов договоров (с индивидуальными шаблонами для каждого университета), автоматическую подстановку данных студентов в формы договоров, а также автоматическую генерацию и подготовку документа к скачиванию для дальнейшей печати.

## **2 Основная часть**

### **2.1 информацию о работе каждого участника в отдельности;**

#### **2.1.1 Аналитик и тим-лид(Матович Стефан)**

На начальном этапе работы над проектом проводился сбор и анализ требований заказчика, результатом чего стала разработка схемы бизнес-процессов, отражающей потребности заинтересованных сторон. На основе этой схемы были сформулированы User Flow, описывающие функциональность подсистемы с точки зрения пользователя. После согласования бизнес-процессов с заказчиком был разработан User Flow. Проведен анализ конкурентов для выявления лучших практик и доработки функциональных требований к подсистеме. Осуществлялось руководство командой, ведение бэклога и заполнение отчетности для контрольных точек.

#### **2.1.2 Дизайнер и разработчик Front-end(ХАНОВ ЭМИЛЬ РУСТЕМОВИЧ)**

На этапе проектирования подсистемы были разработаны варианты интерфейса. С учетом существующего стиля сайта «Точка сбора» был выбран и доработан итоговый дизайн. На этапе разработки оказывалась помощь в верстке и стилистике интерфейса.

#### **2.1.3 Фронтенд-разработчик(Шайхутдинов Кирилл Юрьевич)**

На этапе разработки осуществлялась верстка интерфейса и написание CSS. Проводилось углубленное изучение JavaScript для реализации интерактивных элементов, сложной клиентской логики и взаимодействия с

сервером. Разработаны скрипты, обеспечивающие динамическое отображение данных, проверку пользовательского ввода и отправку данных на сервер. Значительное время было уделено оптимизации JavaScript-кода для повышения производительности и улучшения пользовательского опыта.

#### **2.1.4 Главный Back-end разработчик(Гущин Максим Александрович)**

На этапе разработки осуществлялось проектирование и создание серверной части подсистемы. Проводился поиск оптимальных способов реализации функциональности с использованием микросервисов. Разработана структура базы данных, реализованы бэкенд-компоненты для форм, заполняемых студентами. Осуществлена настройка сервера с использованием nginx.

#### **2.1.5 Второй Back-end разработчик(ЗАРИПОВ ЗАГИР РАИСОВИЧ)**

На этапе разработки ключевой задачей была реализация генератора документов — центрального компонента подсистемы. Осуществлялся выбор оптимальных библиотек и инструментов для создания документов различных форматов (например, PDF, DOCX). Разработаны алгоритмы для динамической подстановки данных студентов в шаблоны документов с учетом требований к форматированию и стилизации. Проводилось тестирование генератора на различных наборах данных и шаблонах, а также оптимизация для обеспечения высокой производительности и масштабируемости. Разработаны механизмы обработки ошибок и логирования, обеспечивающие надежную работу генератора.

## **2.2 разбор требований заказчика и пользователей к программному продукту**

Основные требования заказчика к разрабатываемому программному продукту заключались в создании веб-ориентированного решения для автоматизации процесса оформления приложений к договорам о практике с учебными заведениями. Ключевыми требованиями были:

- 4) Формирование документов: возможность создания печатных форм приложений к договорам с учебными заведениями.
- 5) Сбор данных: Обеспечение заполнения студентами форм с необходимой информацией для оформления практики.
- 6) Гибкость настройки: предоставление организатору практики возможности изменять структуру форм, добавлять новые поля для сбора данных.
- 7) Хранение данных: организация хранения введенных студентами данных в базе данных.
- 8) Автоматическая подстановка данных: обеспечение автоматической подстановки данных студентов из базы данных в итоговые документы;

### **2.3 составление плана действий для достижения цели (backlog);**

В процессе разработки проекта был составлен подробный план действий (бэклог), учитывающий знания, навыки и возможности каждого члена команды. Бэклог включал в себя перечень задач, необходимых для достижения поставленных целей, а также приоритеты и сроки их выполнения. Распределение задач осуществлялось с учётом компетенций участников команды, что позволило оптимизировать процесс разработки и обеспечить эффективное использование ресурсов. В ходе работы бэклог регулярно пересматривался и корректировался с учётом текущего прогресса и возникающих изменений в требованиях.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	
1	Задача	Планируемый результат	Дата начала проекта ->	18.03.2025	18.03.2025	18.03	вт	ср	чт	пт	сб	вс	пн	вт	ср	чт	пт	сб	вс	пн	вт	ср
3	Создать блок-схему анализа ЦА	Созданная блок-схема анализа ЦА в drawIO	Матовин Стефан	23.03.2025	24.03.2025	2																
4	Сделать Схему Бизнес-процессов	Созданная блок-схемы бизнес-процессов в drawIO	Матовин Стефан	25.03.2025	04.04.2025	10																
5	Сделать первый макет сайта	Сделанный один макет в figma	Ханов Эмиль	01.04.2025	08.04.2025	8																
6	Сделать Схему Функционала подсистемы	Созданная блок-схемы функционала подсистемы в drawIO	Матовин Стефан	04.04.2025	11.04.2025	8																
7	Сделать Схему Функционала подсистемы	Созданная блок-схемы функционала подсистемы в drawIO	Матовин Стефан	04.04.2025	11.04.2025	8																
8	Описать логику БД	Описания БД в drawIO	Гущин Максим	05.04.2025	05.04.2025	1																
9	Проектирование Backend	Описание микросервисов на drawIO	Гущин Максим	06.04.2025	13.04.2025	8																
10	сделать второй макет сайта	Сделанный второй макет в figma	Ханов Эмиль	09.04.2025	16.04.2025	8																
11	верстка окна пользователя с html и css	Сверстанный макет окна пользователя	Шайхутдинов Кирилл	11.04.2025	18.04.2025	8																
12	Поднять логинт, расписать БД -> подтяжку с БД	Поднятый логинт, расписанная БД -> миграции в gitхабе	Гущин Максим	13.04.2025	20.04.2025	8																
13	Поднять логинт	Создание скелета приложения на gitхабе	Зарипов Загир	13.04.2025	20.04.2025	8																
14	Сделать юзер стори	Созданный юзер стори в drawIO	Матовин Стефан	13.04.2025	22.04.2025	10																
15	добавить дизайн на макет выбранный макет	Полноценный макет с дизайном	Ханов Эмиль	16.04.2025	23.04.2025	8																
16	верстка окна пользователя с java script	Сверстанный макет пользователя с JS	Шайхутдинов Кирилл	18.04.2025	25.04.2025	8																
17	Написать генератор форм, написать API (rest,	Сделать генерацию форм и заполнить на gitхабе	Гущин Максим	20.04.2025	27.04.2025	8																
18	Расписать БД -> подтяжку с БД -> миграции	Поднятый логинт, расписанная БД -> миграции в gitхабе	Зарипов Загир	20.04.2025	27.04.2025	8																
19	Итерация и оптимизация	Собрать отзывы дизайн и доработать дизайн	Ханов Эмиль	23.04.2025	01.05.2025	9																
20	верстка окна администратора с html и css	Сверстанный макет окна администратора	Шайхутдинов Кирилл	25.04.2025	02.05.2025	8																
21	Помочь команде в реализации фронтэнда и БД	Оказать помощь команде в создании проекта	Матовин Стефан	27.04.2025	06.05.2025	10																
22	Написать генератор	Генератор документов на gitхабе	Зарипов Загир	27.04.2025	04.05.2025	8																
23	Научиться отправлять файл на скачку (ссылку)	получение навыков отправки файлов на скачку, и план реализации	Гущин Максим	27.04.2025	05.05.2025	9																
24	верстка окна администратора с java script	Сверстанный макет окна администратора с JS	Шайхутдинов Кирилл	02.05.2025	09.05.2025	8																
25	Написать API	созданный API на gitхабе	Зарипов Загир	04.05.2025	11.05.2025	8																
26	Подготовить презентацию	созданная презентация и речь для презентации	Матовин Стефан	06.05.2025	13.05.2025	8																
27	интеграция фронтенда и бэкенда	полноценный сайт с фронтендом и бэкендом	Шайхутдинов Кирилл	09.05.2025	18.05.2025	10																
28	Развернуть докер и локальные переменные	Создание докера и пост на gitхаб	Гущин Максим	11.05.2025	18.05.2025	8																
29	Развернуть докер и локальные переменные	развернутый докер в gitхабе	Зарипов Загир	11.05.2025	18.05.2025	8																
30	Подготовить защиту и отчет	созданный отчет и подготовленная защита	Матовин Стефан	13.05.2025	22.05.2025	10																
31	Развернуть докер компос	Создание докера и пост на gitхаб	Гущин Максим	18.05.2025	25.05.2025	8																
32	Помочь в разворачивании приложения	Связанный полностью проект	Зарипов Загир	18.05.2025	25.05.2025	8																
33	Развернуть приложение с фронтендом	Связанный полностью проект	Гущин Максим	18.05.2025	25.05.2025	8																
34	интеграция фронтенда и бэкенда	полноценный сайт с фронтендом и бэкендом	Шайхутдинов Кирилл	09.05.2025	18.05.2025	10																
35																						
36																						

Рисунок 1 – Бэклог в виде таблицы Ганта

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	
1	Задача	Планируемый результат	Дата начала проекта ->	18.03.2025	18.03.2025	18.03	вт	ср	чт	пт	сб	вс	пн	вт	ср	чт	пт	сб	вс	пн	вт	ср
24	верстка окна администратора с java script	Сверстанный макет окна администратора с JS	Шайхутдинов Кирилл	02.05.2025	09.05.2025	8																
25	Написать API	созданный API на gitхабе	Зарипов Загир	04.05.2025	11.05.2025	8																
26	Подготовить презентацию	созданная презентация и речь для презентации	Матовин Стефан	06.05.2025	13.05.2025	8																
27	интеграция фронтенда и бэкенда	полноценный сайт с фронтендом и бэкендом	Шайхутдинов Кирилл	09.05.2025	18.05.2025	10																
28	Развернуть докер и локальные переменные	Создание докера и пост на gitхаб	Гущин Максим	11.05.2025	18.05.2025	8																
29	Развернуть докер и локальные переменные	развернутый докер в gitхабе	Зарипов Загир	11.05.2025	18.05.2025	8																
30	Подготовить защиту и отчет	созданный отчет и подготовленная защита	Матовин Стефан	13.05.2025	22.05.2025	10																
31	Развернуть докер компос	Создание докера и пост на gitхаб	Гущин Максим	18.05.2025	25.05.2025	8																
32	Помочь в разворачивании приложения	Связанный полностью проект	Зарипов Загир	18.05.2025	25.05.2025	8																
33	Развернуть приложение с фронтендом	Связанный полностью проект	Гущин Максим	18.05.2025	25.05.2025	8																
34	интеграция фронтенда и бэкенда	полноценный сайт с фронтендом и бэкендом	Шайхутдинов Кирилл	09.05.2025	18.05.2025	10																
35																						
36																						

Рисунок 2 – Бэклог в виде таблицы Ганта

Поскольку разрабатываемый продукт является составной частью более крупной системы и не представляет собой независимый проект, прямое сравнение с масштабными аналогами нецелесообразно. Основным конкурентом является другая команда, работающая над аналогичным функционалом. Ключевыми преимуществами разрабатываемого решения являются:

#### 2.4.1 Управление руководителями практики:

добавлена возможность добавления руководителей практики организаторами, что позволяет студентам выбирать их из списка, избегая ошибок при вводе ФИО и снижая нагрузку на проверку документов.

#### **2.4.2 Гибкость настройки форм:**

реализована возможность редактирования формы для студентов индивидуально для каждого университета, что позволяет учитывать различные требования к данным, предъявляемые разными учебными заведениями.

### **2.5 обзор архитектуры программного продукта, описание основных компонентов и связей между ними, обоснование выбора архитектурного решения**

Проект реализован в виде трех независимых микросервисов, взаимодействующих между собой для обеспечения функциональности системы:

#### **2.5.1 Сервис аутентификации**

отвечает за вход администраторов в систему. Для авторизации используется REST API, а для проверки авторизации при взаимодействии с другими сервисами — GRPC. Сервис написан на языке Go, что обусловлено опытом разработчика. Тестирование проводилось с использованием стандартной библиотеки Go и МОСК-объектов. Для отслеживания работы сервиса реализовано логирование с помощью стандартной библиотеки SLOG. Выбор библиотек обусловлен их популярностью, наличием обширной документации и простотой взаимодействия.

#### **2.5.2 Сервис заявок**

позволяет создавать, редактировать и отправлять формы для учебных заведений. Для взаимодействия с внешним миром используется REST API, а для взаимодействия с сервисом генерации — GRPC. Сервис также написан на

языке Go по тем же причинам, что и сервис аутентификации. Протестирован с использованием стандартной библиотеки Go и МОСК-объектов, а также реализован логирование через SLOG.

### **2.5.3 Сервис генерации**

отвечает за генерацию документов. Написан на языке Python, что обусловлено опытом разработчика.

### **2.5.4 Технологии и архитектура:**

- 1) Микросервисов связаны между собой посредством GRPC, что обеспечивает высокую скорость и надежность соединения.
- 2) Взаимодействие с пользователями осуществляется через REST API.
- 3) Для балансировки нагрузки и маршрутизации запросов между сервисами используется NGINX.
- 4) Данные хранятся в базе данных PostgreSQL, выбор которой обусловлен её популярностью и достаточной производительностью для ожидаемой нагрузки (до 10 тысяч запросов в секунду);

### **2.5.5 Причины выбора технологий**

Использование Go и Python в качестве основных языков разработки обусловлено опытом разработчиков и их экспертизой в данных технологиях. Применение стандартных и популярных библиотек обеспечивает доступность документации и простоту взаимодействия. GRPC обеспечивает эффективное взаимодействие между микросервисами, а REST API — удобное взаимодействие с пользователями. PostgreSQL является надежным и проверенным решением для хранения данных.

## **2.6 описание методологии разработки и информация о процессе**

## **разработки**

### **2.6.1 описание методологии разработки**

В процессе разработки проекта применялась гибкая методология, основанная на принципах Agile. Использовался итеративный подход, при котором разработка осуществлялась короткими циклами (спринтами), позволяющими быстро реагировать на изменения требований и получать обратную связь от заказчика. В рамках каждого спрингта проводились планирование, разработка, тестирование и демонстрация результатов. Для управления проектом и отслеживания задач использовались инструменты управления проектами. Регулярно проводились командные встречи для обсуждения прогресса, выявления проблем и координации действий. Такой подход позволил обеспечить гибкость, прозрачность и высокую скорость разработки.

### **2.6.2 информация о процессе разработки:**

- 1) **Планирование:** На каждом спрингте проводилось планирование, включающее определение задач, оценку трудозатрат и распределение задач между членами команды.
- 2) **Разработка:** Реализация запланированных задач выполнялась в соответствии с поставленными требованиями.
- 3) **Тестирование:** проводилось модульное и интеграционное тестирование, направленное на выявление и исправление ошибок.
- 4) **Демонстрация:** По завершении каждого спрингта проводилась демонстрация результатов заказчику, обеспечивающая обратную связь и возможность оперативной корректировки плана.
- 5) **Ретроспектива:** по завершении каждого спрингта проводилась ретроспектива, позволяющая проанализировать процесс работы, выявить улучшения и внести изменения в методику разработки;

## **2.7 отчет о результатах тестирования на промежуточных этапах, разбор выявленных ошибок**

В процессе разработки на каждом промежуточном этапе проводилось тестирование разработанного функционала. Тестирование включало:

- 1) Модульное тестирование:** проверка отдельных компонентов системы на соответствие требованиям.
- 2) Интеграционное тестирование:** проверка взаимодействия между различными компонентами системы.
- 3) Пользовательское тестирование:** проверка удобства и понятности интерфейса для конечных пользователей.

Примеры выявленных ошибок и способы их устранения:

**Ошибка:**

при загрузке сгенерированного документа возникала ошибка.

**Решение:**

Процесс генерации и скачивания был отложен, выявлены и устраниены проблемы с формированием файла на сервере и отправкой его клиенту.

**Ошибка:**

в сгенерированном документе отображалась только одна строка данных вместо всех заполненных студентом.

**Решение:**

Исправлена логика подстановки данных в шаблон документа. Проведена проверка корректности цикла обхода данных студентов и правильности вывода информации в соответствующие поля документа.

По результатам тестирования составлялись отчеты, в которых фиксировались выявленные ошибки, их приоритетность и способы устранения. Устранение ошибок осуществлялось в рамках текущего или следующего спринта. Регулярное тестирование на промежуточных этапах позволило своевременно выявлять и устранять проблемы, обеспечивая высокое качество разрабатываемого продукта.

## **2.8 информация о планировании деятельности в ходе разработки и распределении задач между участниками команды разработчиков.**

Планирование деятельности в рамках проекта осуществлялось на каждом спринте с использованием гибкой методологии Agile. Задачи определялись на основе требований заказчика и анализа пользовательских историй. Учитывая опыт и компетенции каждого члена команды, задачи распределялись таким образом, чтобы максимально эффективно использовать имеющиеся ресурсы.

Для управления задачами и отслеживания прогресса был создан и поддерживался в актуальном состоянии backlog. Backlog содержал подробное описание каждой задачи, ее приоритет, оценку трудозатрат и ответственного за ее выполнение. Регулярно проводились командные встречи, на которых обсуждался прогресс, выявлялись проблемы и корректировался план действий.

Такой подход к планированию и распределению задач позволил обеспечить прозрачность процесса разработки, эффективную координацию действий и своевременное достижение поставленных целей.

## **3 ЗАКЛЮЧЕНИЕ**

### **3.1 Оценка соответствия программного продукта поставленным требованиям заказчика**

Разработанный программный продукт был оценен на предмет соответствия требованиям заказчика и пользователей. Анализ показал, что реализованный функционал в полной мере отвечает ключевым потребностям, заявленным в начале проекта.

В частности, реализованы следующие важные аспекты:

- 1) Автоматизация формирования документов:** подсистема обеспечивает автоматическую генерацию документов с данными студентов, что значительно экономит время организаторов практики.
- 2) Гибкость настройки:** реализована возможность настройки форм для каждого университета, что позволяет учитывать специфические требования к данным.
- 3) Удобство использования:** интерфейс системы разработан с учетом удобства пользователей, что позволяет студентам и организаторам легко ориентироваться и выполнять необходимые действия.
- 4) Надежность и стабильность:** проведенное тестирование показало стабильную работу системы и отсутствие критических ошибок.

Таким образом, можно сделать вывод, что разработанный программный продукт успешно соответствует требованиям заказчика и обеспечивает удобное и эффективное решение для автоматизации процесса оформления документов о прохождении практики.

## **3.2 оценка качества программного продукта на основе результатов тестирования**

Несмотря на наличие незначительных недостатков, в целом разработанный программный продукт можно оценить как качественный и готовый к эксплуатации. Своевременное выявление и устранение ошибок в процессе тестирования позволило обеспечить стабильную и надежную работу системы, а также соответствие требованиям заказчика и пользователей.

## **3.3 Вывод**

Успешно разработана подсистема автоматизации оформления документов о практике, отвечающая требованиям заказчика и пользователей, реализован ключевой функционал, включающий автоматическую генерацию документов, гибкую настройку форм и удобный интерфейс, обеспечена стабильная и надежная работа системы, подтвержденная результатами тестирования. Применение гибкой методологии Agile позволило оперативно реагировать на изменения требований и обеспечило высокую скорость разработки, эффективное планирование и распределение задач между членами команды способствовало оптимальному использованию ресурсов, систематическое тестирование на промежуточных этапах позволило своевременно выявлять и устранять ошибки. Рекомендуется продолжать использовать гибкие методологии разработки для обеспечения адаптивности к изменяющимся требованиям, уделять особое внимание тестированию на ранних этапах разработки для своевременного выявления и устранения ошибок, при планировании проекта учитывать возможность масштабирования системы и расширения функционала в будущем, совершенствовать коммуникацию внутри команды и с заказчиком для лучшего понимания требований и эффективного решения проблем.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Документация Go. Язык программирования Go. —  
URL: <https://go.dev/doc/> (дата обращения: 20.04.2025).
2. gRPC. Официальный сайт gRPC. — URL: <https://grpc.io/> (дата обращения: 20.05.2025).
3. Документация Python. Официальный сайт Python. —  
URL: <https://www.python.org/doc/> (дата обращения: 06.04.2025).
4. slog. Пакет slog стандартной библиотеки Go. —  
URL: <https://pkg.go.dev/log/slog> (дата обращения: 18.04.2025).
5. MOCK. Библиотека для имитации в Go. — URL: (Укажите конкретную библиотеку, если она использовалась, например, <https://github.com/golang/mock>) (дата обращения: 19.04.2025).
6. REST API. Определение и концепции. — URL: (укажите источник, объясняющий основы REST API, например, <https://www.redhat.com/en/topics/api/what-is-a-rest-api>) (дата обращения: 28.03.2025).
7. NGINX. Официальный сайт NGINX. —  
URL: <https://www.nginx.com/> (дата обращения: 21.04.2025).
8. Документация PostgreSQL. Официальный сайт PostgreSQL. —  
URL: <https://www.postgresql.org/docs/> (дата обращения: 1.04.2025).
9. Веб-фреймворк Gin. Быстрый веб-фреймворк на основе HTTP. Написан на Go. — URL: <https://github.com/gin-gonic/gin> (дата обращения: 4.04.2025)
10. Документация по JavaScript. Сеть разработчиков Mozilla (MDN). —  
URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (дата обращения: 1.05.2025)
11. Документация Docker. Официальный сайт Docker. —

URL: <https://docs.docker.com/> (дата обращения: 4.05.2025)

12. Нильсен, Дж. Инженерия удобства использования. — Морган Кауфманн, 1993 (дата обращения: 28.03.2025)

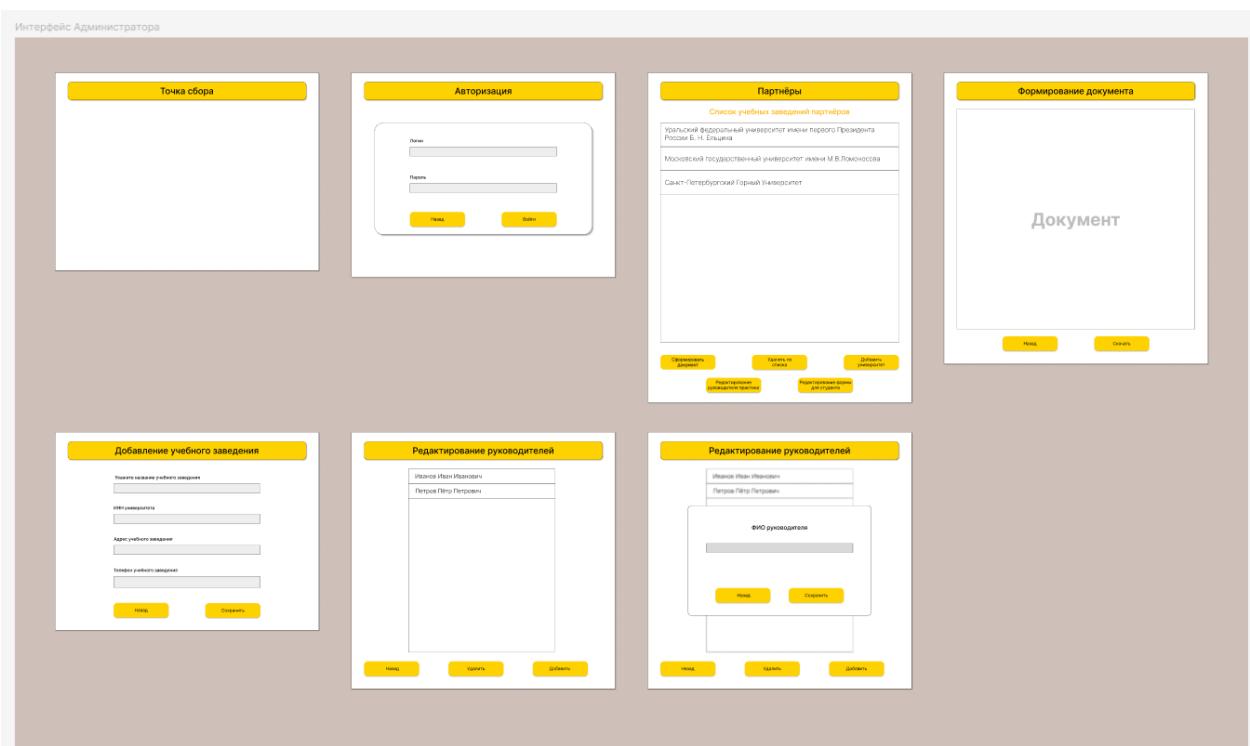
## **ПРИЛОЖЕНИЕ А**

### **BACKLOG**

Календарный план представлен в отдельном документе. Ознакомиться с ним можно по следующей ссылке:  
<https://docs.google.com/spreadsheets/d/1qTN2fdcdDmLmUdcSvRr75vIgNd36UO-/edit?usp=sharing&ouid=105384476983349023412&rtpof=true&sd=true> (дата обращения: 26.05.2025).

## **ПРИЛОЖЕНИЕ В**

### **FIGMA**



Интерфейс студента

The image shows a user interface for a student application. At the top left, the text "Интерфейс студента" is displayed. Below it, there are two main sections:

- Точка сбора**: A large, empty white rectangular area.
- Подача заявления**: A form with the following fields:
  - ФИО студента
  - Выберите университет
  - Сроки практики
  - Руководитель практики
  - Тип практики

At the bottom right of the "Подача заявления" section, there are two buttons: "Назад" (Back) and "Сохранить" (Save).