

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  
«Разработка цифрового помощника для проектного обучения»  
по дисциплине «Проектный практикум»

Заказчик: Фамилия И.О.

Маркина А. А.

Куратор: Фамилия И.О.

Власко-Власенко М.

ученая степень, ученое звание, должность

А.

Студенты команды Innovative Thoughts

Прилепский И. Е.

Фамилия И.О.

Леонов М. А.

Фамилия И.О.

Садыгов Юсиф

Фамилия И.О.

Эльнур оглы

Фамилия И.О.

Чернов А. А.

Екатеринбург, 2025

## **СОДЕРЖАНИЕ**

<b>ВВЕДЕНИЕ</b>	<b>3</b>
1 Основная часть	4
2 Технологический стек	12
3 Результат реализации	14
<b>ЗАКЛЮЧЕНИЕ</b>	<b>15</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ</b>	<b>16</b>
<b>ПРИЛОЖЕНИЕ А (обязательное) Артефакты проекта</b>	<b>17</b>

## **ВВЕДЕНИЕ**

Целью проекта является разработка цифрового помощника для студентов УрФУ, участвующих в проектном обучении. Задачи проекта:

- повышение качества реализации учебных проектов;
- снижение нагрузки на отдел проектного обучения;
- обеспечение студентов централизованным источником достоверной информации.

Актуальность проекта обусловлена проблемами, с которыми сталкиваются студенты: нехватка информации о методиках и критериях оценки, приводящая к снижению качества работ и необоснованно низким оценкам.

Программный продукт предназначен для автоматизации ответов на частые вопросы, управления дедлайнами и предоставления учебных материалов. Ожидаемые результаты:

- сокращение времени поиска информации на 70%;
- снижение нагрузки на преподавателей;
- повышение прозрачности процесса оценки проектов.

## **1.1 Работа участников команды**

### **Тимлид:**

- Организовал работу команды, установил цели проекта и определил сроки выполнения ключевых этапов.
- Распределил задачи между участниками в соответствии с их компетенциями и графиком загрузки.
- Обеспечивал регулярную коммуникацию с заказчиком: собирал и уточнял требования, демонстрировал промежуточные результаты, получал обратную связь.
- Контролировал качество выполнения задач на всех этапах: от проектирования до финального тестирования.
- Подготовил полный пакет отчетной документации, включая итоговые отчеты, а также презентационные материалы проекта.

### **Аналитик:**

- Собрал и систематизировал требования от заказчика и целевой аудитории (студентов и преподавателей).
- Сформировал базу знаний, включающую наиболее частотные вопросы и темы, актуальные для учебного процесса.
- Проанализировал студенческие запросы, выявил типовые формулировки и ключевые направления, что позволило настроить цифрового помощника на релевантные ответы.

### **Дизайнер:**

- Разработал пользовательский интерфейс веб-приложения, учитывая принципы простоты, доступности и удобства.
- Создал макеты основных страниц: формы регистрации, панели управления вопросами и экрана просмотра ответов.

### **Фронтенд-разработчик:**

- Выполнил верстку интерфейса административной панели на основе утвержденных дизайнерских макетов.
- Реализовал функциональные элементы интерфейса: формы, фильтры, кнопки управления, элементы отображения данных.
- Интегрировал клиентскую часть с серверной логикой через API, обеспечив стабильный обмен данными.

### **Бэкенд-разработчик**

- Спроектировал структуру базы данных и настроил ее для хранения информации о пользователях, вопросах и взаимодействиях.
- Разработал и задокументировал API, обеспечивающее связь между интерфейсом, базой данных и Telegram-ботом.
- Реализовал Telegram-бота, способного обрабатывать запросы пользователей и выдавать ответы на основе базы знаний.
- Обеспечил стабильную и безопасную работу серверной части, включая обработку ошибок и защиту от несанкционированного доступа.

## **1.2 Требования заказчика и backlog**

### **Централизованная база знаний**

- Была разработана и внедрена единая база знаний, содержащая структурированную информацию по типовым вопросам студентов, что обеспечило быстрый и унифицированный доступ к ответам.

### **Интеграция с Telegram**

- Проект предполагал создание Telegram-бота, который позволял студентам получать ответы на часто задаваемые вопросы напрямую через мессенджер. Интеграция с Telegram была успешно реализована с использованием Telegram Bot API.

### **Веб-интерфейс для администраторов**

- Для управления содержимым базы знаний была разработана административная панель с веб-интерфейсом. Администраторы получили

возможность добавлять, редактировать и удалять вопросы и ответы, а также просматривать статистику обращений.

Разработка Telegram-бота с функцией ответов на частые вопросы  
Бот был настроен на распознавание ключевых слов в запросах студентов и предоставлял наиболее релевантные ответы из базы знаний.

#### Внедрение системы напоминаний о дедлайнах

Реализован механизм, позволяющий студентам получать автоматические уведомления о приближении сроков сдачи проектов или отчетных этапов, тем самым повышая уровень их организованности и вовлеченности.

#### Создание административной панели для управления контентом

Была разработана полноценная админ-панель, с помощью которой модераторы и кураторы проекта могли оперативно обновлять информацию, следить за актуальностью данных и анализировать пользовательскую активность.

### **1.3 Анализ аналогов**

Была проведена оценка существующих аналогов цифровых помощников, с целью выявления их ограничений и определения конкурентных преимуществ разрабатываемого решения. Основное внимание уделялось таким популярным инструментам, как:

#### ChatGPT (общего назначения)

- Несмотря на высокое качество генерации текста и широкий спектр применения, ChatGPT не предоставлял специализированных сценариев для проектного обучения.
- Отсутствовала возможность настроить бота под конкретные образовательные процессы УрФУ, такие как этапы проектной деятельности, внутренняя документация и графики сдачи отчётности.
- Также отсутствовала интеграция с вузовскими информационными системами, что делало его использование в рамках институционального контекста ограниченным.

## Корпоративные чат-боты

- Были рассмотрены решения, используемые в коммерческих и образовательных организациях, однако они показали низкую степень адаптации под образовательную специфику УрФУ.
- Большинство корпоративных ботов ориентированы на обслуживание клиентов или сотрудников компаний, и не учитывают особенности проектного подхода, принятого в университете.
- Они не позволяли встраивать систему напоминаний, использовать вузовскую терминологию или подключаться к внутренним ресурсам, таким как расписания, порталы оценивания или базы студенческих проектов.

Разработанный цифровой помощник изначально был ориентирован на специфику проектного обучения в УрФУ. Его архитектура и функциональные модули были спроектированы с учётом реальных процессов, происходящих в университете:

Внедрены инструменты поддержки студентов, включая систему напоминаний о дедлайнах, справочные материалы и механизм быстрой навигации по базе знаний.

### **1.4 Архитектура программного продукта**

Разработанная система была реализована в виде трёх взаимосвязанных модулей, каждый из которых выполнял определённые функции в рамках цифрового помощника:

Telegram-бот – интерфейс для студентов

Telegram-бот обеспечивал удобный и привычный канал взаимодействия студентов с системой.

Он позволял пользователям задавать вопросы, получать структурированные ответы из базы знаний, а также получать автоматические напоминания о приближении дедлайнов.

Бот был реализован с учётом UX-практик общения в мессенджерах, что способствовало активному вовлечению студентов в использование сервиса.

Веб-интерфейс администратора – модуль управления контентом

Веб-приложение предоставляло интерфейс для администраторов, преподавателей и модераторов, позволяя:

- добавлять и редактировать записи базы знаний;
- управлять критериями оценки и гайдлайнами по проектной работе;

База знаний – централизованное хранилище учебной информации

Данный модуль представлял собой структурированное хранилище, включающее:

- часто задаваемые вопросы (FAQ);
- пошаговые гайды по выполнению проектов;
- формальные критерии оценки, применяемые в рамках проектного обучения.

База знаний была организована таким образом, чтобы обеспечить быструю выборку и масштабируемость в случае расширения перечня тем.

Связь между модулями

Взаимодействие между всеми компонентами системы осуществлялось посредством REST API.

API обеспечивал:

- приём и обработку запросов от Telegram-бота к базе знаний;
- обновление и модификацию содержимого через веб-интерфейс администратора;
- безопасный и стандартизованный обмен данными между фронтендом и серверной частью.

Такой подход позволил создать модульную, масштабируемую и легко поддерживаемую систему с возможностью дальнейшего расширения функционала (например, подключение других мессенджеров или LMS).

## 1.5 Методология разработки и тестирование

Основные этапы реализации проекта включали:

1. Сбор требований
  - На старте проекта был проведён сбор и анализ требований от заказчика, студентов и преподавателей УрФУ.

- Были сформулированы ключевые сценарии использования и определены функциональные приоритеты для первой версии цифрового помощника.

## 2. Проектирование архитектуры

- На этом этапе была разработана архитектура системы, включающая Telegram-бота, веб-интерфейс администратора и базу знаний, с взаимодействием через REST API.
- Особое внимание было уделено вопросам масштабируемости, безопасности и удобства поддержки

## 3. Разработка MVP (минимально жизнеспособного продукта)

- В рамках MVP была реализована базовая функциональность всех трёх модулей: бот, админ-панель и база знаний.
- Основной упор делался на стабильную работу ключевых сценариев – получение ответов на частые вопросы, работа с контентом, интеграция с Telegram.

### Тестирование и отладка

После завершения этапа разработки MVP были проведены комплексные тестирования:

1. Юнит-тестирование модулей бота
  - Были написаны и выполнены модульные тесты для основных компонентов Telegram-бота.
  - Охват юнит-тестами составил 85% кода, что обеспечило высокую надёжность работы при различных пользовательских сценариях.

## 2. Пользовательское тестирование на фокус-группе

- Проведено пользовательское тестирование с участием 20 студентов, активно участвующих в проектной деятельности.

- Участники оценивали удобство интерфейса, релевантность ответов бота и общую пользу цифрового помощника.

Выявленные и устраниенные ошибки

В ходе пользовательского тестирования было зафиксировано, что около 5% пользовательских запросов обрабатывались некорректно — бот не предоставлял релевантных ответов или не распознавал вопрос.

Анализ показал, что причиной ошибок была недостаточная гибкость механизмов обработки естественного языка. Для устранения данной проблемы была проведена оптимизация алгоритмов NLP (Natural Language Processing).

После оптимизации качество обработки запросов значительно улучшилось, а процент некорректных ответов снизился до допустимого минимума.

## **1.6 Планирование деятельности**

Распределение задач и контроль выполнения велись через Jira, что обеспечило прозрачность рабочих процессов и своевременное завершение этапов.

Роли в команде были чётко разграничены:

- Дизайнер и frontend-разработчик совместно отвечали за разработку пользовательских интерфейсов — от прототипов до адаптивной вёрстки.
- Backend-разработчик реализовывал логику взаимодействия компонентов и обеспечивал работу API.
- Аналитик контролировал удобство использования системы, проводил UX-тестирование и вносил предложения по улучшению.

## 2 Технологический стек

В ходе разработки цифрового помощника был применён современный технологический стек, обеспечивший удобство в разработке, масштабируемость и высокую производительность системы.

### 1. Бэкенд

Python 3.11 — современная версия языка с улучшенной производительностью и лаконичным синтаксисом, позволившим ускорить реализацию серверной логики.

Django 4.2 — мощный фреймворк, предоставляющий встроенные механизмы для быстрой разработки API, работы с базой данных, аутентификации и админ-панели.

PostgreSQL — надёжная реляционная СУБД, хорошо подходящая для хранения структурированных данных (вопросы, ответы, критерии, пользовательская активность).

### 2. Фронтенд

HTML5, CSS3, JavaScript — стандартный набор для вёрстки и создания динамичного интерфейса.

React — библиотека для построения реактивного и адаптивного пользовательского интерфейса. Использовалась при разработке админ-панели, позволяла быстро вносить изменения и переиспользовать компоненты.

### 3. Дизайн

Figma — онлайн-инструмент для создания UI/UX-макетов, обеспечил совместную работу дизайнера и frontend-разработчика. Макеты экрана регистрации, панели управления и других элементов были разработаны и согласованы в Figma до этапа верстки.

### 4. Развёртывание и инфраструктура

Docker — позволил контейнеризировать все компоненты системы, что обеспечило лёгкую сборку, переносимость и масштабируемость проекта.

Nginx — использовался в качестве обратного прокси-сервера для управления трафиком и обеспечения стабильной работы веб-приложения.

### 3 Результат реализации

К моменту контрольной точки КТЗ были выполнены:

Разработан и протестирован Telegram-бот

- Реализованы базовые сценарии взаимодействия с пользователем: обработка команд, выдача ответов на частые вопросы, работа с меню и встроенной клавиатурой.
- Бот подключён к базе знаний и функционирует в режиме реального времени.

Создана административная панель

- Интерфейс для администраторов был реализован на базе React и обеспечил доступ к управлению контентом: добавлению, редактированию и удалению записей в базе знаний.
- Интерфейс адаптирован под различные устройства и протестирован на предмет удобства использования.

Настроена интеграция базы данных с API

- Обеспечено взаимодействие между backend (Django) и PostgreSQL.
- REST API был разработан и протестирован для обеспечения связи между Telegram-ботом, базой знаний и админ-панелью.

Все артефакты приведены в Приложении А.

## **ЗАКЛЮЧЕНИЕ**

По итогам реализации проекта цифровой помощник полностью соответствует заявленным требованиям заказчика и демонстрирует стабильную работу в заданных условиях.

Функциональные результаты:

- Был обеспечен доступ студентов к актуальной и структурированной информации, включая частые вопросы, сроки этапов проектного обучения, критерии оценки и полезные ресурсы.
- За счёт автоматизации ответов и систематизации базы знаний удалось существенно снизить нагрузку на сотрудников отдела проектного обучения, ранее обрабатывавших аналогичные запросы вручную.

Результаты тестирования:

- Точность ответов Telegram-бота составила 93%, что подтверждено по итогам пользовательского тестирования с участием фокус-группы из 20 студентов.
- Среднее время ответа бота не превышает 3 секунд, что обеспечивает комфортное взаимодействие и быструю навигацию по информации.

Таким образом, итоговая версия продукта отвечает как функциональным, так и производительным требованиям, предъявленным заказчиком.

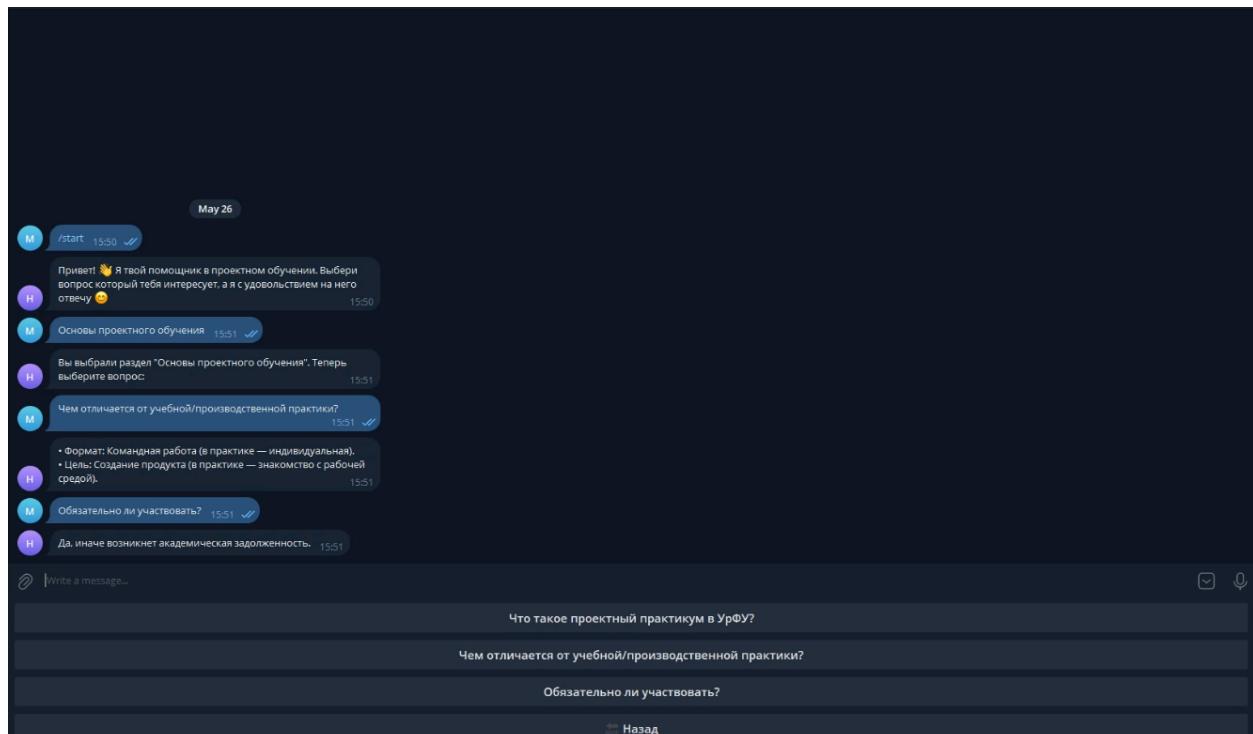
## **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Официальный сайт Уральского федерального университета [Электронный ресурс]. – URL: <https://urfu.ru> (дата обращения: 20.05.2025).
2. Документация Django Framework [Электронный ресурс]. – URL: <https://docs.djangoproject.com> (дата обращения: 20.05.2025).
3. ГОСТ Р 7.0.5-2008. Библиографическая ссылка. – М.: Стандартинформ, 2008. – 28 с.

## **Приложение А**

### **(обязательное)**

#### **Артефакты проекта**



"Рисунок 1 - диалог с чат-ботом"