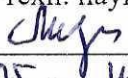


Министерство науки и высшего образования Российской Федерации
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Институт прикладной математики и компьютерных наук

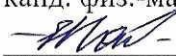
ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК
Руководитель ОПОП
д-р техн. наук, профессор
 К.И. Лившиц
« 05 » июнь 2023 г.


ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

ЧАТ-БОТ ДЛЯ IT-СОБЕСЕДОВАНИЙ В TELEGRAM,
ИСПОЛЬЗУЮЩИЙ НЕЙРОННУЮ СЕТЬ

по направлению подготовки 01.03.02 Прикладная математика и информатика,
направленность (профиль) «Прикладная математика и информатика»

Овсянникова Ксения Александровна

Руководитель ВКР
канд. физ.-мат. наук, доцент
 Е.Г.Пахомова
« 01 » июня 2023 г.

Автор работы
студент группы № 931921
 К.А.Овсянникова
« 01 » июня 2023 г.

Министерство науки и высшего образования Российской Федерации.
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)
Институт прикладной математики и компьютерных наук

УТВЕРЖДАЮ
Руководитель ОПОП
д-р техн. наук, профессор

 А.М. Горцев

подпись
«21» ноября 2022 г.

ЗАДАНИЕ

по выполнению выпускной квалификационной работы бакалавра обучающегося
Овсянниковой Ксении Александровне

Фамилия Имя Отчество обучающегося

по направлению подготовки 01.03.02 Прикладная математика и информатика,
направленность (профиль) «Прикладная математика и информатика»

1 Тема выпускной квалификационной работы

Чат-бот для IT-собеседований в Telegram, использующий нейронную сеть

2 Срок сдачи обучающимся выполненной выпускной квалификационной работы:

а) в учебный офис / деканат – 01.06.2023 б) в ГЭК – 07.06.2023

3 Исходные данные к работе:

Объект исследования – Чат-бот в Telegram

Предмет исследования – Telegram-бот, API

Цель исследования – Реализация чат-бота для IT-собеседований в Telegram

Задачи:

Провести обзор и сравнительный анализ средств для разработки сервиса. Разработать архитектуру Telegram-бота. Реализовать чат-бот в Telegram.

Методы исследования:

Использование языка программирования Python и библиотеки python-telegram-bot для реализации чат-бота в Telegram

Организация или отрасль, по тематике которой выполняется работа, –
НИ ТГУ, Институт прикладной математики и компьютерных наук,
кафедра компьютерной безопасности

4 Краткое содержание работы

Обзор и сравнительный анализ средств для разработки сервиса, проектирование архитектуры чат-бота, практическая реализация чат-бота

Руководитель выпускной

квалификационной работы

Кандидат физ.-мат.наук, доцент, ТГУ

должность, место работы



подпись

Е.Г.Пахомова

И.О. Фамилия

Задание принял к исполнению

Студентка группы 931921, ТГУ

должность, место работы



подпись

К.А.Овсянникова

И.О. Фамилия

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
1 Обзор и сравнительный анализ средств для разработки сервиса	6
1.1 Общие понятия чат-бота. Функции чат-бота. Классификация чат-ботов.....	6
1.2 Выбор программного обеспечения для разработки чат-бота.....	10
1.3 Преимущества применения чат-ботов в контексте IT-собеседований	13
2 Архитектура чат-бота для IT-собеседований в Telegram.....	15
2.1 Telegram-бот API	15
2.2 Проектирование архитектуры чат-бота.....	16
2.3 Функция ограничения времени на вопрос	18
3 Практическая реализация чат-бота	19
3.1 Реализация функциональности опросника на примере IT-компании Digital Nomads	19
3.2 Связь нейронной сети (ChatGPT) с чат ботом для IT-собеседований.....	21
3.3 Возникшие проблемы и их решение.....	24
3.4 Результаты работы Telegram-бота	27
ЗАКЛЮЧЕНИЕ.....	29
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ	31

ВВЕДЕНИЕ

В современном мире информационные технологии прочно вошли во все сферы жизни и стали неотъемлемой частью нашего общества. Быстрый темп развития технологий приводит к тому, что в сфере IT постоянно возникают новые задачи и требования, требующие наличия квалифицированных специалистов. Однако поиск и отбор подходящих кандидатов на IT-позиции является сложной и трудоемкой задачей, которая требует значительных временных и финансовых затрат.

Традиционные методы проведения собеседований, основанные на личном присутствии кандидата и вопросах от рекрутера, могут быть неэффективными в оценке навыков и компетенций кандидатов. Кроме того, они требуют значительных затрат времени и ресурсов, особенно при большом потоке кандидатов. В свете этих проблем, использование чат-ботов для проведения IT-собеседований становится все более популярным и востребованным подходом.

Целью данной дипломной работы является разработка и реализация чат-бота для IT-собеседований в популярном мессенджере Telegram. Основная задача чат-бота состоит в проведении опросника для кандидатов с ограничением времени на ответы, которое устанавливается динамически в зависимости от сложности вопроса. Такой подход позволит создать реалистичное собеседование, где кандидатам придется демонстрировать свои знания и навыки в ограниченное время, а рекрутерам предоставится возможность более эффективно оценить их профессиональные качества.

Однако простое проведение опроса не всегда достаточно для полноценной оценки кандидатов. Для повышения качества и эффективности собеседования, в работе предусмотрена интеграция чат-бота с нейронной сетью, особенно с использованием платформы ChatGPT. Связь с нейронной сетью позволит чат-боту обрабатывать и анализировать ответы кандидатов, идентифицировать ключевые навыки, а также предоставлять кандидатам обратную связь и рекомендации для дальнейшего развития.

В рамках работы будут рассмотрены существующие научные и практические подходы к созданию чат-ботов и проведению IT-собеседований. Будет проанализирован опыт различных компаний и рекрутинговых агентств в использовании чат-ботов при отборе кандидатов, а также изучены существующие решения в области разработки чат-ботов и использования нейронной сети.

Важной частью работы будет тестирование и оценка эффективности разработанного чат-бота. Будут проведены различные эксперименты и сравнения с традиционными методами собеседований, чтобы убедиться в эффективности и надежности чат-бота.

В заключении будут подведены итоги работы, оценены достигнутые результаты и выдвинуты предложения по дальнейшему совершенствованию и расширению функциональности разработанного чат-бота для IT-собеседований.

Таким образом, данная дипломная работа имеет практическую значимость и направлена на создание инновационного инструмента для проведения IT-собеседований, что поможет существенно оптимизировать процесс найма квалифицированных IT-специалистов и повысить эффективность отбора кандидатов.

1 Обзор и сравнительный анализ средств для разработки сервиса

1.1 Общие понятия чат-бота. Функции чат-бота. Классификация чат-ботов

Чат-боты представляют собой интеллектуальные программы, способные выполнять различные функции и задачи. Они находятся внутри мессенджеров, таких как Telegram, Facebook Messenger или WhatsApp, и могут быть доступными для пользователей по запросу или автоматически активироваться при определенных условиях.

Чат-боты – это уникальные аккаунты, которые функционируют в качестве виртуальных собеседников, не привязанных к конкретному человеку. Они обрабатывают и анализируют сообщения, отправленные с них или адресованные им, с помощью внешней системы. Однако для пользователя взаимодействие с чат-ботом выглядит так же, как обычное общение с реальным человеком.

Функциональность чат-ботов может быть разнообразной. Некоторые чат-боты предоставляют информацию о продуктах или услугах, отвечают на вопросы пользователей или помогают совершать покупки. Другие могут выполнять функции переводчиков, рассылать новости или даже играть в игры.

Особенностью чат-ботов является их способность общаться с пользователями на естественном языке. Они используют алгоритмы обработки естественного языка и машинного обучения, чтобы понимать вопросы и команды пользователей, а затем предоставлять соответствующие ответы или выполнять запрошенные действия.

Чат-боты могут выполнять разнообразные функции, которые зависят от их конкретного предназначения и целей. Вот некоторые из наиболее распространенных функций чат-ботов:

1. Предоставление информации. Чат-боты могут быть созданы для предоставления различных типов информации. Это может включать ответы на вопросы пользователей, предоставление информации о продуктах или услугах, актуальных новостей, расписаний и других справочных данных.

2. Консультации и поддержка. Чат-боты могут помогать пользователям в получении консультаций или поддержке в определенной области. Например, чат-боты в сфере медицины могут предоставлять рекомендации по здоровью и лечению, а чат-боты в сфере бизнеса могут консультировать пользователей по разным вопросам.

3. Работа с заказами и бронированиями. Чат-боты могут обрабатывать заказы и запросы на бронирование ресурсов от пользователей. Например, чат-боты могут помочь в заказе товаров или услуг, бронировании отелей, ресторанов, авиабилетов и т.д. Они могут принимать запросы, обрабатывать платежи и отправлять подтверждения пользователю.

4. Интерактивные игры и развлечения. Чат-боты могут предлагать различные игры и развлекательные активности для пользователей. Например, это могут быть текстовые квесты, головоломки, викторины или даже ролевые игры. Чат-боты в этом случае выступают в роли виртуальных персонажей или проводят пользователей через интерактивные истории.

5. Автоматизация задач. Чат-боты могут выполнять автоматизацию различных задач, которые обычно требуют взаимодействия с реальным человеком. Например, они могут помогать в заполнении форм, сборе информации от пользователей, создании и обновлении профилей, отправке уведомлений и многое другое.

6. Интеграция с внешними сервисами. Чат-боты могут интегрироваться с различными внешними сервисами и приложениями для выполнения специфических функций. Например, они могут интегрироваться с платежными системами для обработки платежей, с почтовыми сервисами для отправки уведомлений или с CRM-системами для управления клиентскими данными или как в моем случае с ChatGPT для обработки ответов пользователя.

Это лишь некоторые примеры функций, которые могут быть реализованы в чат-ботах. Фактически, возможности чат-ботов ограничены только техническими и креативными возможностями разработчиков. Они могут быть настроены для выполнения почти любой задачи, которую можно автоматизировать или упростить с помощью коммуникации с компьютерной программой.

Чат-боты могут быть классифицированы по различным критериям. Вот некоторые из наиболее распространенных классификаций чат-ботов:

1. По уровню сложности и функциональности.

- **Простые/правила-основанные чат-боты.** Эти чат-боты основаны на заранее заданных правилах и шаблонах ответов. Они могут предоставлять ограниченный набор ответов на основе predetermined правил и шаблонов.

- Искусственно-интеллектуальные (ИИ) чат-боты: Эти чат-боты используют алгоритмы машинного обучения и искусственного интеллекта для понимания и генерации ответов. Они способны обрабатывать более сложные запросы и взаимодействовать с пользователями на более естественном уровне.

2. По типу взаимодействия.

- Текстовые чат-боты – это наиболее распространенный тип чат-ботов, которые взаимодействуют с пользователями через текстовые сообщения. Они могут быть интегрированы в мессенджеры, социальные сети или веб-интерфейсы.
- Голосовые чат-боты – эти чат-боты используют голосовое распознавание и синтез для взаимодействия с пользователями. Они могут быть доступны через голосовых помощников или телефонные системы.

3. По области применения.

- Сервисные чат-боты. Эти чат-боты предоставляют сервисы и информацию пользователю. Например, это могут быть банковские чат-боты, туристические ассистенты или помощники по онлайн-покупкам.
- Развлекательные чат-боты. Эти чат-боты созданы для развлечения пользователей. Они могут предлагать игры, шутки, головоломки или истории.

Это лишь некоторые примеры классификаций чат-ботов. В реальности, чат-боты могут сочетать различные характеристики и быть уникальными в своем роде. Чат-боты становятся все более популярными и востребованными в различных областях, включая бизнес, образование, медицину, развлечения и IT-сферу. Они предоставляют новые возможности для автоматизации задач, улучшают сервис, позволяют автоматизировать процессы коммуникации, сократить время ожидания ответов и повысить удобство для пользователей.

На рисунке 1 представлен график популярности чат-ботов Telegram в разных сферах деятельности.

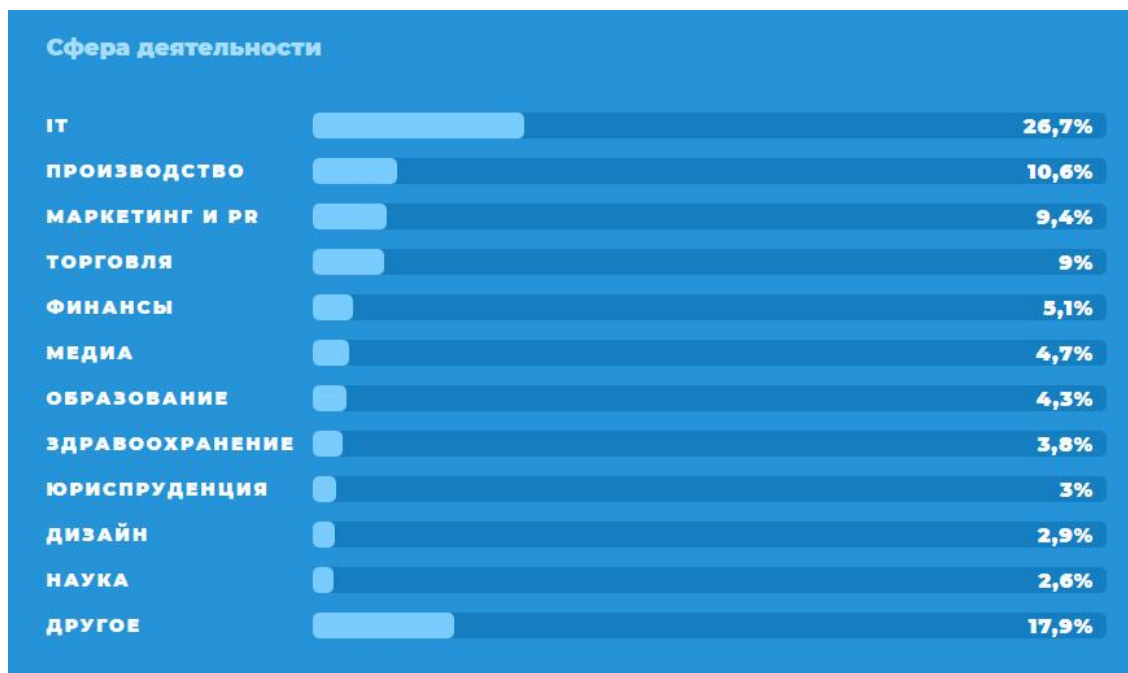


Рисунок 1 – График популярности чат-ботов Telegram в разных сферах деятельности

1.2 Выбор программного обеспечения для разработки чат-бота

Для разработки чат-ботов в Telegram существует несколько вариантов программного обеспечения, которые предоставляют удобные инструменты и API для создания и управления ботами. Вот некоторые из них:

1. **Telegram Bot API.** Telegram предоставляет свой собственный API для разработки чат-ботов. Он предлагает набор методов и возможностей для взаимодействия с ботами, включая отправку сообщений, обработку входящих запросов и управление настройками бота. Этот API доступен на различных языках программирования и предоставляет широкие возможности для создания чат-ботов.

2. **Botpress.** Botpress – это открытая платформа для разработки чат-ботов с поддержкой Telegram и других платформ мессенджеров. Он предлагает графический интерфейс и набор инструментов для создания, обучения и управления ботами. Botpress поддерживает различные языки программирования и предоставляет API для интеграции с другими системами.

3. **Microsoft Bot Framework.** Microsoft Bot Framework предоставляет инструменты и платформу для разработки различных типов чат-ботов, включая ботов для Telegram. Он поддерживает несколько языков программирования, таких как C#, Node.js и Python, и предлагает набор SDK, библиотек и сервисов для создания и развертывания ботов.

4. **Dialogflow.** Dialogflow (ранее известный как API.AI) - это инструмент разработки чат-ботов, который предлагает распознавание и понимание естественного языка (NLU) и функциональность машинного обучения. Dialogflow поддерживает интеграцию с Telegram и другими мессенджерами, предоставляя возможности создания разговорных интерфейсов и обработки пользовательских запросов.

5. **Rasa.** Rasa – это фреймворк с открытым исходным кодом для разработки и обучения чат-ботов с поддержкой Telegram и других платформ. Он предоставляет набор инструментов для создания разговорной логики, обработки естественного языка и обучения моделей машинного обучения. Rasa позволяет создавать гибких и контекстно-ориентированных чат-ботов.

Для разработки чат-бота в Telegram одним из наиболее популярных вариантов программного обеспечения является использование языка программирования Python. Python – это мощный и простой в использовании язык, который обладает обширной экосистемой и богатым набором инструментов для разработки чат-ботов.

Основные преимущества использования Python для разработки чат-ботов в Telegram:

Простота и интуитивность. Python имеет чистый и понятный синтаксис, что делает его привлекательным для начинающих разработчиков. Это позволяет быстро и легко создавать прототипы и разрабатывать функциональность бота.

Поддержка Telegram API. Существуют Python-библиотеки, которые облегчают взаимодействие с Telegram API, такие как `python-telegram-bot` и `telebot`. Они предоставляют высокоуровневые абстракции и простой интерфейс для работы с API Telegram, позволяя разработчикам легко создавать и управлять чат-ботами.

Широкие возможности интеграции. Python предлагает множество библиотек для работы с искусственным интеллектом (AI) и машинным обучением (ML), таких как TensorFlow, PyTorch, и scikit-learn. Это открывает двери для интеграции с функциями ИИ в чат-боте и создания умных и автоматизированных систем.

Для разработки чат-бота на Python в Telegram, была выбрана библиотека: `python-telegram-bot`. И были изучены документация и примеры использования этой библиотеки. Эта библиотека предоставляет удобные методы для обработки входящих сообщений, управления клавиатурой, отправки и получения данных через Telegram API.

В целом, использование Python для разработки чат-бота в Telegram предоставляет широкие возможности и преимущества, позволяя создать гибкого и функционального бота с минимальными затратами времени и усилий.

На рисунке 2 показан рейтинг языков 2023 года в коммерческих рабочих проектах.






Dec 2022	Dec 2021	Change	Programming Language	Ratings	Change
1	1		 Python	16.66%	+3.76%
2	2		 C	16.56%	+4.77%
3	4	▲	 C++	11.94%	+4.21%
4	3	▼	 Java	11.82%	+1.70%
5	5		 C#	4.92%	-1.48%
6	6		 Visual Basic	3.94%	-1.46%
7	7		 JavaScript	3.19%	+0.90%
8	9	▲	 SQL	2.22%	+0.43%
9	8	▼	 Assembly language	1.87%	-0.38%
10	12	▲	 PHP	1.62%	+0.12%

Рисунок 2 – Рейтинг языков программирования 2023 года в коммерческих рабочих процессах.

По данным, приведенным на рисунке 2, можно сделать вывод, что Python значительно опережает JavaScript и сейчас является самым популярным языком программирования. В пятерку лучших языков вошли также: Go, Java, TypeScript.

На рисунке 3 представлен рейтинг популярности языков по личным предпочтениям в %.

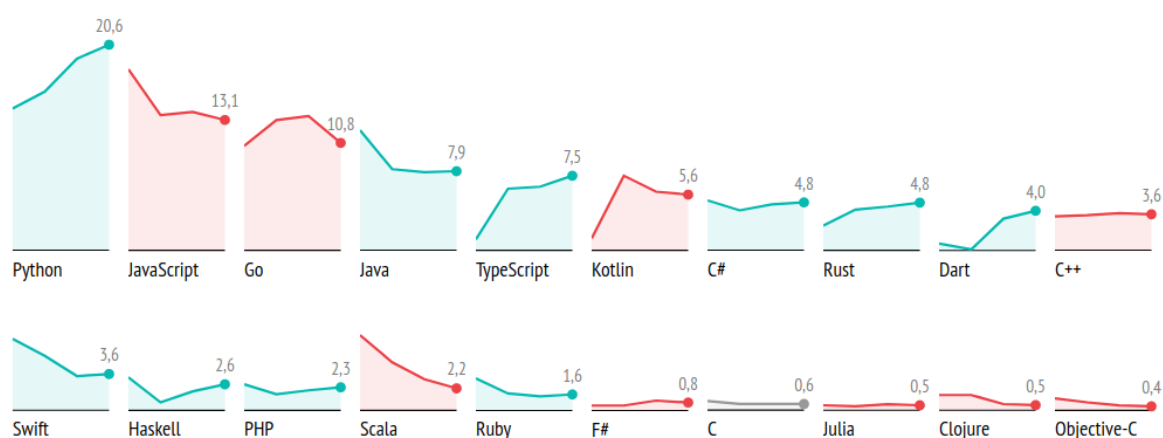


Рисунок 3 – Рейтинг популярности языков по личным предпочтениям в %.

1.3 Преимущества применения чат-ботов в контексте IT-собеседований

Применение чат-ботов в контексте IT-собеседований предлагает ряд преимуществ, которые могут значительно облегчить и улучшить процесс подбора кандидатов. Вот некоторые из них:

- **Эффективность и скорость.** Чат-боты позволяют автоматизировать процесс проведения собеседований, что позволяет значительно сэкономить время и ресурсы. Кандидаты могут проходить опросник и отвечать на вопросы бота в удобное для них время, а результаты могут быть мгновенно обработаны. Это повышает удобство и доступность процесса для кандидатов.

- **Объективность и согласованность.** Чат-боты предлагают стандартизированный подход к проведению собеседований, обеспечивая одинаковый набор вопросов и критериев оценки для всех кандидатов. Это помогает исключить субъективность и предвзятость, обеспечивая более объективную оценку навыков и компетенций.

- **Гибкость и настраиваемость.** Чат-боты позволяют гибко настраивать опросники и вопросы в соответствии с требованиями и потребностями компании. Время, отведенное на ответы кандидата, может быть установлено динамически в зависимости от сложности вопроса или предоставленного времени.

- **Масштабируемость.** Чат-боты позволяют проводить собеседования с несколькими кандидатами одновременно, что позволяет существенно увеличить масштаб подбора персонала. Боты способны обслуживать большое количество пользователей одновременно без потери качества и эффективности.

- **Улучшенная аналитика и отчетность.** Чат-боты могут предоставлять детальные аналитические данные о процессе собеседований, такие как результаты ответов кандидатов, время, затраченное на каждый вопрос и т.д. Это позволяет более глубоко анализировать эффективность процесса и принимать обоснованные решения по отбору кандидатов.

- **Связь с нейросетью.** Чат-боты могут быть интегрированы с системами искусственного интеллекта (ИИ), такими как ChatGPT, для более продвинутой обработки и анализа ответов кандидатов. ИИ может помочь в автоматическом выделении ключевых навыков и оценке качества ответов.

Применение чат-ботов в IT-собеседованиях предлагает множество преимуществ, повышая эффективность, объективность и удобство процесса подбора кандидатов. Это инновационное решение, которое помогает компаниям сократить затраты на ресурсы и повысить качество найма IT-специалистов.

2 Архитектура чат-бота для IT-собеседований в Telegram

2.1 Telegram-бот API

Это обычный HTTP-interface для взаимодействия с чат-ботами в мессенджере Telegram. Каждый созданный бот должен уметь автоматически отправлять сообщения и принимать входящие. Для этого нам и нужен сервер, принимающий HTTP запросы и выдающий ответы.

Есть всего 2 способа получать обновления с сервера:

1. **long pulling** – телеграмм сам будет раз в какое-то время, например в 1 секунду, спрашивать сервер о новых уведомлениях;
2. **webhook** – сервера сами будут присылать оповещение, сразу как появится новое уведомление.

Все оповещения будут сохранены на сервере без обработки, но не долго, максимум до 24 часов. В ответ будет отправлен объект «Апдейт», заранее сериализованный в JSON.

Все запросы к Telegram-боту API осуществляются через HTTPS в виде: https://api.telegram.org/bot/НАЗВАНИЕ_МЕТОДА. Принцип взаимодействия бота и пользователя изображен на рисунке 4.

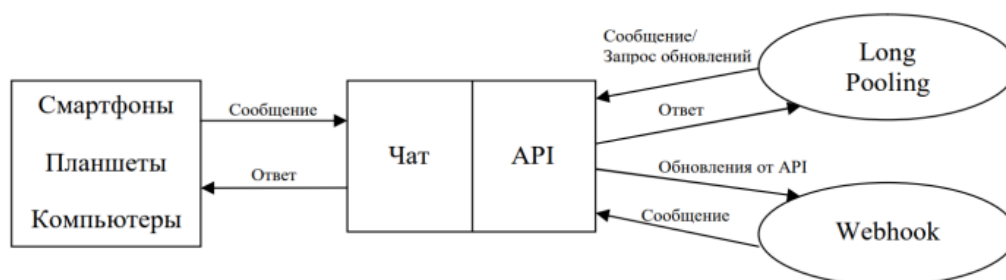


Рисунок 4 – Принцип взаимодействия бота и пользователя.

Чтобы получить для бота ключ доступа, он же токен, необходимо написать botfather боту. Он создан для того, чтобы создавать ботов. Ниже приведены некоторые примеры методов:

1. **getUpdates** – этот метод используется для получения обновлений по технологии long polling;
2. **sendMessage** – метод осуществляет обычную отправку сообщения пользователю.

2.2 Проектирование архитектуры чат-бота

На рисунке 5 представлены файлы, используемые в данной дипломной работе для чат-бота в Telegram:

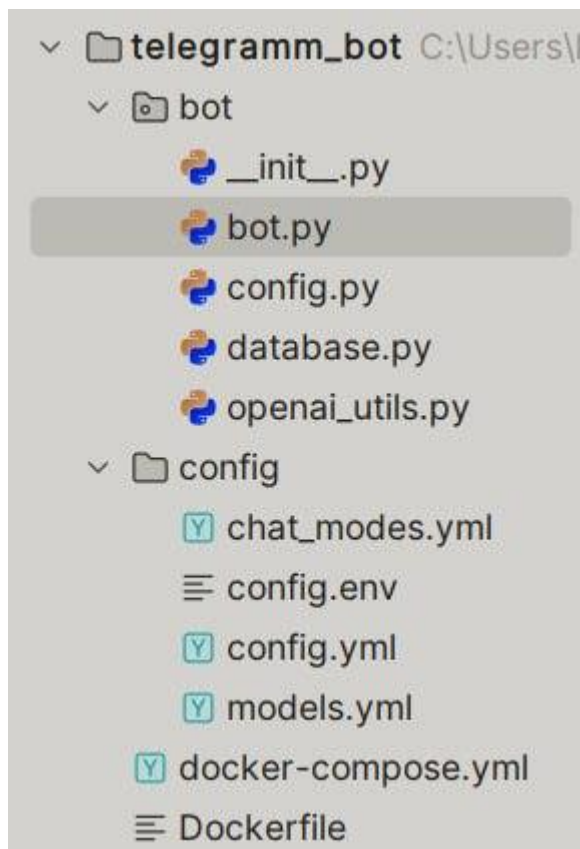


Рисунок 5 – Файлы чат-бота в Telegram

Bot.py – главный файл. В нем определено стартовое окно, меню бота, обработка исключений, создание новых диалогов, отправка сообщений, добавление новых пользователей. Другими словами, все взаимодействие с пользователем.

Config.py – это файл с настройками бота. Здесь устанавливаются токены: telegram-токен, ChatGPT-токен, chatGPT-токен, настройки таймера и тд.

Database.py – это файл базы данных. В нем хранится информация о пользователях: ID, chat Id, Имя, дата регистрации, дата последнего обращения, получение и отправка сообщений и токен пользователя.

Openai_utils.py – интеграция ChatGPT. Взаимодействие с сервисом ChatGPT.

Папка config – имеет конфигурационные файлы, такие как: mode chat (меню),

токен telegram бота, API Key сервиса ChatGPT, доступны (логин, пароль, порт и путь volums). Содержит информацию, которую нужно вывести при нажатии на кнопку. Например, «пройти собеседование». Также, в эти файлы записываются токены и версии.

На рисунке 6 приведена схема взаимодействия пользователя с чат-ботом Telegram.

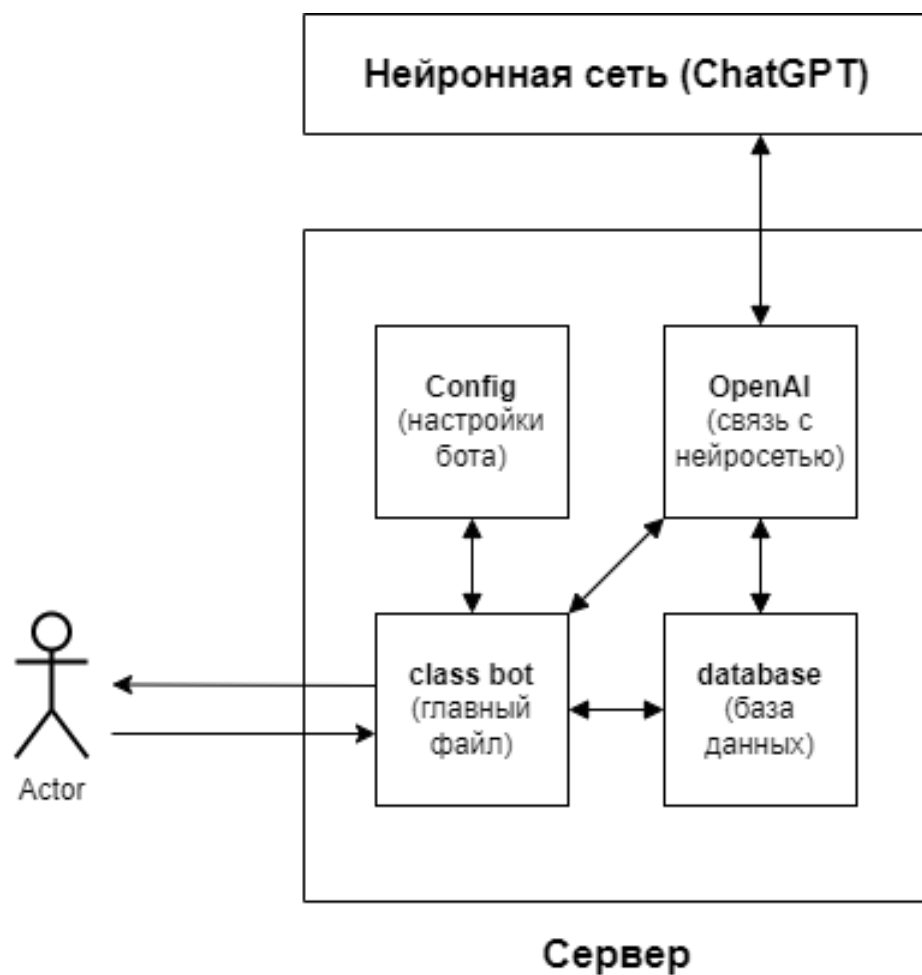


Рисунок 6 – Схема взаимодействия пользователя с чат-ботом Telegram.

2.3 Функция ограничения времени на вопрос

Для того, чтобы процесс собеседования IT-специалистов был честным и отслеживаемым HR-менеджерами, была разработана функция вывода таймера в чат-бот. Эта функция запускает счетчик времени, который был заранее определен на каждый вопрос. Если собеседуемый специалист не успевает вовремя ответить на вопрос, то вопрос не зачитывает. Бот выводит следующий вопрос в диалоговое окно Telegram, без возможности продолжать ответ на предыдущий вопрос.

Реализация функции счетчика времени в коде представлена на рисунке 7.

```
ync def message_handle_fn():
    # new dialog timeout
    if use_new_dialog_timeout:
        if (datetime.now() - db.get_user_attribute(user_id, "last_interaction")).seconds > config.new_dialog_timeout and len(db.get_dialog_messages(user_id)) > 0:
            db.start_new_dialog(user_id)
            await update.message.reply_text(f"Заканчивается новое окно из-за timeout (<b>{config.chat_modes[chat_mode]['name']}</b> mode) ❌", parse_mode=ParseMode.HTML)
            db.set_user_attribute(user_id, "last_interaction", datetime.now())

    # in case of CanceledError
    n_input_tokens, n_output_tokens = 0, 0
    current_model = db.get_user_attribute(user_id, "current_model")
```

Рисунок 7 – Реализация счетчика времени в коде.

Время для каждого вопроса было определено заранее. Если вопрос подразумевает счетчик времени, то в коде вызывается данная функция, изображенная на рисунке 8.

```
new_dialog_timeout = config_yaml["new_dialog_timeout"]
```

Рисунок 8 – Функция вызова счетчика времени.

3 Практическая реализация чат-бота

3.1 Реализация функциональности опросника на примере IT-компаний Digital Nomads

Для реализации Telegram-бота была проделана следующая работа.

1. Создали бота и получили токен через @BotFather:

- Зарегистрировались в мессенджере Telegram.
- Нашли в Telegram бота @BotFather и запустили его.
- Следуя инструкциям @BotFather, создали нового бота и получили уникальный токен для доступа к API.

2. Проектирование и выбор стека технологий для реализации backend-части бота:

- Определили функциональные требования к боту и его возможности.
- Выбрали язык программирования и фреймворк для backend-разработки бота, учитывая требования проекта и собственные навыки.
- Рассмотрели различные инструменты и технологии, такие как базы данных, веб-серверы, архитектурные шаблоны и прочее, для поддержки функциональности бота.

3. Регистрация и выдача API Key на сервисе OpenAI:

- Зарегистрировались на сайте OpenAI и создали аккаунт.
- Получили API Key, который будет использоваться для взаимодействия с сервисом OpenAI.

4. Реализация backend-части.

а) Связь с Telegram (с помощью библиотеки python-telegram-bot):

- Установили библиотеку python-telegram-bot для работы с Telegram Bot API в выбранном языке программирования.

- Использовали полученный ранее токен бота для настройки соединения с Telegram API.
- Определили обработчики событий и команд бота, такие как прием сообщений, обработка команд, ответы на запросы и прочее.

б) Связь с сервисом OpenAI:

- Использовали API Key, полученный от OpenAI, для настройки соединения с их сервисом.
- Определили функции и методы, которые будут использоваться для отправки запросов к OpenAI API и получения ответов, сгенерированных моделью ИИ.

в) Контейнеризация (docker-compose) backend:

- Создали Docker-контейнеры, которые содержат необходимое программное обеспечение и зависимости для запуска и работы бота.
- Использовали инструмент docker-compose для управления и настройки контейнеров, а также для определения конфигурации и параметров среды выполнения.

Каждый из этих шагов включает более подробные этапы и требует соответствующих знаний и навыков в области разработки программного обеспечения. Также важно учитывать специфические особенности выбранного языка программирования и используемых инструментов для эффективной реализации функциональности чат-бота в контексте IT-собеседований.

3.2 Связь нейронной сети (ChatGPT) с чат ботом для IT-собеседований

Интеграция нейронной сети, такой как ChatGPT, в чат-бота для IT-собеседований с использованием программного обеспечения Telegram Bot API может значительно усилить возможности бота и обеспечить более эффективное и интерактивное взаимодействие с пользователями. Вот несколько шагов по интеграции нейросети (ChatGPT) в чат-бота, используя Telegram Bot API:

1. **Создание бота в Telegram.** Создаем бота в Telegram и получаем API-ключ, который будет использоваться для взаимодействия с Telegram Bot API.
2. **Установка и настройка ChatGPT.** Устанавливаем и настраиваем ChatGPT для работы с чат-ботом. Это включает в себя создание учетной записи ChatGPT, получение доступа к API и настройку параметров и конфигураций моделей ИИ, которые хотим использовать.
3. **Настройка взаимодействия с Telegram Bot API.** Используем соответствующие библиотеки и фреймворки на языке программирования Python - настройки взаимодействия с Telegram Bot API. Это включает создание обработчиков для различных команд и сообщений, отправляемых пользователем, и настройку логики обработки этих сообщений.
4. **Связь с ChatGPT.** Внедряем функции ChatGPT в чат-бот для обработки запросов и генерации ответов. Это включает использование NLP-моделей ChatGPT для понимания и анализа текстовых запросов пользователей.
5. **Обработка и анализ ответов.** Разрабатываем механизм обработки и анализа ответов, полученных от ChatGPT. Это включает оценку и классификацию ответов, проверку на соответствие заданным критериям.
6. **Тестирование и настройка.** Тестируем и настраиваем чат-бот с интеграцией ChatGPT, чтобы убедиться, что он работает должным образом и предоставляет точный анализ ответов кандидатов.

Связь нейронной сети (ChatGPT) с чат-ботом для IT-собеседований с использованием программного обеспечения Telegram Bot API позволяет боту обрабатывать и анализировать сложные запросы, предоставлять более точный анализ ответов и выставлять оценки.

Ниже представлена реализация класса связи нейронной сети (ChatGPT) с Telegram-ботом в коде программы.

На рисунке 8 представлена функция отправки сообщения на серверы нейронной сети и получение ответа через токен.

```
8 OPENAI_COMPLETION_OPTIONS = {
9     "temperature": 0.7,
10    "max_tokens": 1000,
11    "top_p": 1,
12    "frequency_penalty": 0,
13    "presence_penalty": 0
14 }
15
16
17 1 usage
18 class ChatGPT:
19     def __init__(self, model="gpt-3.5-turbo"):
20         assert model in {"text-davinci-003", "gpt-3.5-turbo", "gpt-4"}, f"Unknown model: {model}"
21         self.model = model
22
23     5 usages (4 dynamic)
24     async def send_message(self, message, dialog_messages=[], chat_mode="assistant"):
25         if chat_mode not in config.chat_modes.keys():
26             raise ValueError(f"Chat mode {chat_mode} is not supported")
27
28         n_dialog_messages_before = len(dialog_messages)
29         answer = None
30         while answer is None:
31             try:
32                 if self.model in {"gpt-3.5-turbo", "gpt-4"}:
33                     messages = self._generate_prompt_messages(message, dialog_messages, chat_mode)
34                     r = await openai.ChatCompletion.acreate(
35                         model=self.model,
36                         messages=messages,
37                         **OPENAI_COMPLETION_OPTIONS
```

Рисунок 8 – Функция отправки сообщения на серверы нейронной сети и получения ответа через токен.

На рисунке 9 представлена функция создания диалогового окна.

```
2 usages
114 def _generate_prompt(self, message, dialog_messages, chat_mode):
115     prompt = config.chat_modes[chat_mode]["prompt_start"]
116     prompt += "\n\n"
117
118     # add chat context
119     if len(dialog_messages) > 0:
120         prompt += "Chat:\n"
121         for dialog_message in dialog_messages:
122             prompt += f"User: {dialog_message['user']}\n"
123             prompt += f"Assistant: {dialog_message['bot']}\n"
124
125     # current message
126     prompt += f"User: {message}\n"
127     prompt += "Assistant: "
128
129     return prompt
130
```

Рисунок 9 – Функция создания диалогового окна

На рисунке 10 представлена функция создания диалогового окна с сообщением от нейронной сети (ChatGPT).

```
2 usages
131 def _generate_prompt_messages(self, message, dialog_messages, chat_mode):
132     prompt = config.chat_modes[chat_mode]["prompt_start"]
133
134     messages = [{"role": "system", "content": prompt}]
135     for dialog_message in dialog_messages:
136         messages.append({"role": "user", "content": dialog_message["user"]})
137         messages.append({"role": "assistant", "content": dialog_message["bot"]})
138     messages.append({"role": "user", "content": message})
139
140     return messages
141
2 usages
142 def _postprocess_answer(self, answer):
143     answer = answer.strip()
144     return answer
145
```

Рисунок 10 – Функция создания диалогового окна с сообщением от нейронной сети (ChatGPT)

На рисунке 11 представлена функция передачи токенов для отправки и приема запросов.

```
1 usage
146 def _count_tokens_from_messages(self, messages, answer, model="gpt-3.5-turbo"):
147     encoding = tiktoken.encoding_for_model(model)
148
149     if model == "gpt-3.5-turbo":
150         tokens_per_message = 4 # every message follows <im_start>{role/name}\n{content}<im_end>\n
151         tokens_per_name = -1 # if there's a name, the role is omitted
152     elif model == "gpt-4":
153         tokens_per_message = 3
154         tokens_per_name = 1
155     else:
156         raise ValueError(f"Unknown model: {model}")
157
```

Рисунок 11 – Функция передачи токенов для отправки и приема запросов

На рисунке 12 представлена реализация перевода аудиосообщения в текстовое и функция генерации изображения .

```
1 usage
183 async def transcribe_audio(audio_file):
184     r = await openai.Audio.atranscribe("whisper-1", audio_file)
185     return r["text"]
186
187
1 usage
188 async def generate_images(prompt, n_images=4):
189     r = await openai.Image.create(prompt=prompt, n=n_images, size="512x512")
190     image_urls = [item.url for item in r.data]
191     return image_urls
192
193
194 async def is_content_acceptable(prompt):
195     r = await openai.Moderation.create(input=prompt)
196     return not all(r.results[0].categories.values())
197
```

Рисунок 12 – Реализация перевода аудиосообщения в текстовое и функция генерации изображения

3.3 Возникшие проблемы и их решение

Для того, чтобы нейронная сеть (ChatGPT) могла обрабатывать ответы, нужно снять сервер за пределами РФ и загрузить на него docker программы.

Проблема возникла во время отладки программы в связи с блокировкой сервера, снятого во Франкфурте. Из-за этого, чат-бот перестал обрабатывать ответы на введенные вопросы.

Чтобы текст ошибки не выводился в диалоговое окно, была разработана функция обработки исключения. На рисунке 13 приведен вариант с отключением данной функции. На рисунке 14 изображен вариант с функцией обработки исключений.

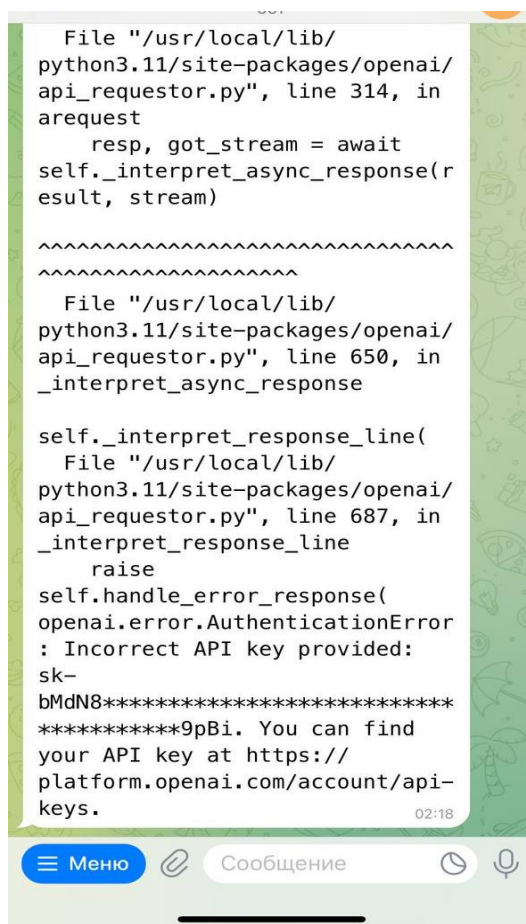


Рисунок 13 – Текст ошибки в диалогом окне из-за отключения функции обработки исключения

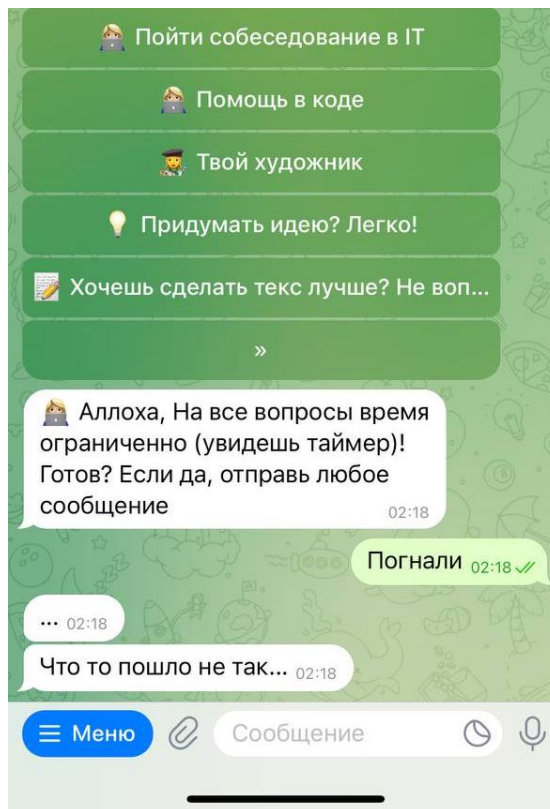


Рисунок 14 – Вывод в диалоговое окно текста из функции исключения

На рисунке 15 показана реализация функции исключения ошибки типа «е» в коде программы с выводом текстового сообщения «Что то пошло не так...».

```
except Exception as e:
    error_text = f"Что то пошло не так..."
    logger.error(error_text)
    await update.message.reply_text(error_text)
    return
```

Рисунок 15 – Функция исключения ошибки типа «е».

Данная проблема решилась снятием другого vds-сервера, арендованного за пределами РФ.

На рисунке 16 приведена правильная работа чат-бота – начало общения пользователя с чат-ботом.

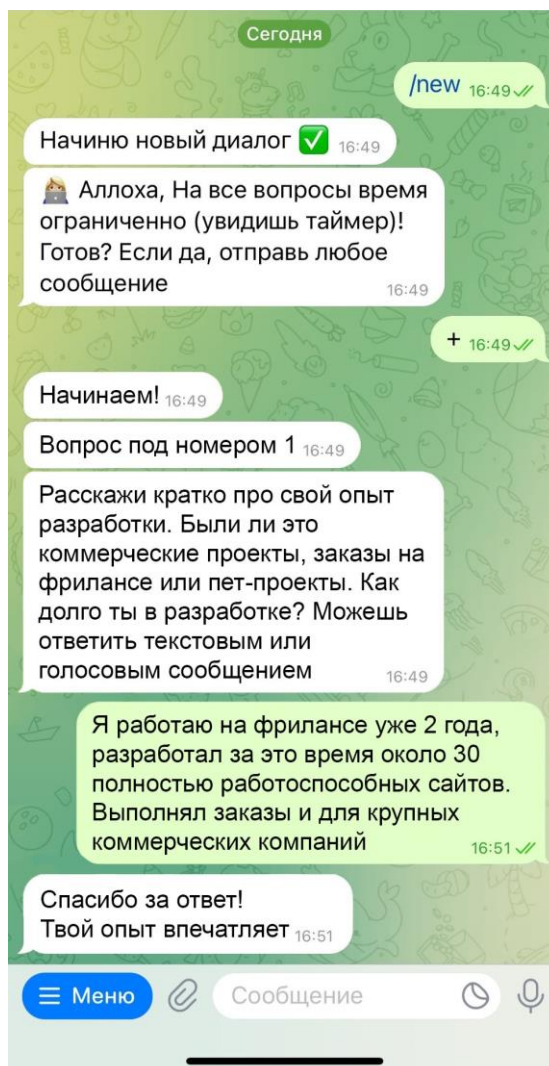


Рисунок 16 – Начало общения пользователя с чат-ботом (работа без ошибок).

3.4 Результаты работы Telegram-бота

После прохождения пользователем чат-бота, результаты его собеседования отправляются HR-менеджеру в виде ссылки: <https://digital-nomads.hrbot.work/interview-results.php?code=CODE-ИНДИВИДУАЛЬНЫЙ-НОМЕР>. Результаты собеседования хранятся в файле программы database.py и записываются в таблицу, которая формируется в bot.py. Таблица с результатами хранится на том же сервере, что и программа самого бота.

Пример таблицы результатов изображен на рисунке 17. Первый столбец – задание. Второй столбец – вопрос к заданию. Третий столбец – ответ, предоставленный пользователем. Четвертый столбец – ответ, ожидаемый от пользователя (правильный ответ). Пятый столбец – время, потраченное на решение задания.

<pre>e = 1 request_number: int er = value t(value) l() } var a: A? struct B { var b: (B)? } struct C { var c: C? =</pre>	<p>Что будет выведено в консоль?</p>	<p>Ошибка получения значения value</p>	<p>Ошибка получения значения value</p>	<p>8 м. 39 с.</p>
<pre>var a: A? struct B { var b: (B)? } struct C { var c: C? =</pre>	<p>Скомпилируется ли код?</p>	<p>Ошибка в C</p>	<p>Ошибка в C</p>	<p>2 м. 50 с.</p>
<pre>=</pre>	<p>Что такое Optional?</p>	<p>Enum</p>	<p>Enum</p>	<p>4 с.</p>
<pre>var a: A? var b: B? var c: C? var d: D? var e: E? var f: F? var g: G? var h: H? var i: I? var j: J? var k: K? var l: L? var m: M? var n: N? var o: O? var p: P? var q: Q? var r: R? var s: S? var t: T? var u: U? var v: V? var w: W? var x: X? var y: Y? var z: Z?</pre>	<p>Что будет выведено в консоль?</p>	<p>true/true</p>	<p>true/true</p>	<p>2 м. 21 с.</p>
<pre>var a: A? var b: B? var c: C? var d: D? var e: E? var f: F? var g: G? var h: H? var i: I? var j: J? var k: K? var l: L? var m: M? var n: N? var o: O? var p: P? var q: Q? var r: R? var s: S? var t: T? var u: U? var v: V? var w: W? var x: X? var y: Y? var z: Z?</pre>	<p>После каких print(#) сработают наблюдатели свойств?</p>	<p>2,3,4,6</p>	<p>2,3,4,6</p>	<p>11 м. 3 с.</p>

Рисунок 17 – Фрагмент таблицы результатов собеседования

Для визуализации правильности ответов были использованы цветные заливки третьего столбца в таблице результатов. В зависимости от цвета, HR-менеджер понимает какую ошибку допустил пользователь.

Значение использования цветной заливки в таблице:

- Зеленый – ответ пользователя полностью совпадает с ожидаемым ответом (правильным ответом), заложенным в программу;
- Желтый – ответ пользователя неверен, т.е. не совпадает с ответом, который ожидает получить программа;

- Красный – ответ пользователя не был предоставлен вовремя. Счетчик времени, данного на вопрос истек.

В верхней части таблицы располагается имя пользователя и его стек технологий, а также время начала собеседования и его окончания. На рисунке 18 изображено начало таблицы результатов.

Максим Физик (iOS)				
10 March 11:39:49 - 10 March 12:19:15 (MSK)				
Изображение	Вопрос	Ответ	Должно быть	Время ответа

Рисунок 18 – Верхняя часть таблицы результатов с именем пользователя, его стеком технологий и временем начала собеседования и окончания

Под таблицей с результатами собеседования отмечается сколько правильных ответов было предоставлено пользователем. На рисунке 19 можно ознакомиться с этим примером.

-	Какова сложность сортировки пузырьком?	O(1)	Активация Windows Чтобы активировать Windows, перейдите в раздел "Параметры".	1 м. 16 с.
---	--	------	--	------------

Максим Физик 2/17

Рисунок 19 – Вывод количества правильных ответов, предоставленных пользователем.

ЗАКЛЮЧЕНИЕ

В данной дипломной работе была представлена разработка и реализация чат-бота для IT-собеседований в популярном мессенджере Telegram. Основная цель работы заключалась в создании опросника для кандидатов с ограничением времени на ответы, которое устанавливается динамически в зависимости от сложности вопроса, а также в интеграции с искусственным интеллектом (ИИ) путем использования платформы ChatGPT. Это позволило создать эффективный инструмент для отбора и оценки кандидатов на IT-позиции.

В процессе работы были достигнуты следующие результаты. Во-первых, был разработан и реализован чат-бот, способный проводить опрос кандидатов, предлагая им вопросы с ограничением времени на ответы. Динамическое установление времени позволило создать реалистичную ситуацию, в которой кандидатам приходится проявлять свои навыки и знания в ограниченное время. Это позволило рекрутерам более эффективно оценить компетенции и профессиональные качества кандидатов.

Во-вторых, была осуществлена интеграция чат-бота с платформой ChatGPT, которая позволила использовать мощные алгоритмы искусственного интеллекта для обработки и анализа ответов кандидатов. Это дало возможность автоматической и более объективной оценки ответов, выявлению ключевых навыков и предоставлению кандидатам обратной связи и рекомендаций. Интеграция с ИИ значительно повысила качество и эффективность собеседований, а также уменьшила нагрузку на рекрутеров.

Тестирование разработанного чат-бота показало его высокую эффективность и точность в оценке кандидатов. Были проведены эксперименты и сравнения с традиционными методами собеседований, которые подтвердили преимущества и надежность чат-бота в процессе отбора кандидатов на IT-позиции.

Однако, несмотря на достигнутые результаты, данная дипломная работа оставляет место для дальнейшего развития и совершенствования. Важным направлением для будущих исследований является расширение функциональности чат-бота, добавление новых типов вопросов и возможностей для анализа ответов кандидатов. Также стоит улучшить процесс интеграции с ИИ, чтобы повысить его точность и адаптивность к различным типам вопросов и ситуациям.

В заключение можно сказать, что разработанный чат-бот для IT-собеседований в

Telegram с реализацией опросника для кандидатов с ограничением времени на вопросы и интеграцией с ИИ является инновационным и эффективным инструментом для отбора квалифицированных IT-специалистов. Он позволяет существенно сократить время и ресурсы, затрачиваемые на проведение собеседований, а также повысить качество и объективность процесса отбора. Результаты работы могут быть полезны для компаний и рекрутинговых агентств, занимающихся подбором IT-персонала, а также для дальнейших исследований в области разработки чат-ботов и применения ИИ в рекрутинге.

Полный код приложения можно посмотреть по ссылке:
<https://github.com/kovsyannikova/---.git> .

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ И ЛИТЕРАТУРЫ

1. Регламент обработки обращений клиентов в чате [Электронный ресурс]. – URL: <https://wiki.skbkontur.ru/pages/viewpage.action?pageId=313263686> (дата обращения: 01.06.2023)
2. Понятие языка программирования [Электронный ресурс]. – URL: <https://ibrain.kz/informatika/ponyatie-yazyk-programmirovaniya> (дата обращения: 01.06.2023)
3. Язык программирования Python [Электронный ресурс]. – URL: <https://web-creator.ru/articles/python> (дата обращения: 01.06.2023)
4. Чат-боты – кто они и что умеют? [Электронный ресурс]. – URL: <https://efsol.ru/articles/messendzhery-i-chat-boty-dlya-biznesadostavki.html> (дата обращения: 01.06.2023)
5. Классификация и методы создания чат-бот приложений [Электронный ресурс]. – URL: <https://cyberleninka.ru/article/n/klassifikatsiya-i-metodysozdaniya-chat-bot-prilozheniy/viewer> (дата обращения: 01.06.2023)
6. Чем отличаются чат-боты в Telegram, WhatsApp, Viber, Facebook, VK [Электронный ресурс]. – URL: <https://www.rpv-bot.ru/chem-otlichaetsyachat-bot-v-telegram-whatsapp-vk-viber-facebook> (дата обращения: 01.06.2023)
7. Самоучитель PYTHON [Электронный ресурс]. – URL: <http://pythoshka.ru/p1138.html> (дата обращения: 01.06.2023)
8. Telegram Bot API Documentation: – URL: <https://core.telegram.org/bots/api> (дата обращения: 01.06.2023)
9. ChatGPT API Documentation: – URL: <https://docs.ChatGPT.com/api/> (дата обращения: 01.06.2023)



ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ

«ДИДЖИТАЛ НОМАДС»

634057, ТОМСКАЯ область, г. ТОМСК, ул. ГОВОРОВА, дом 46/1, кв. 115
Томский Филиал АО "АЛЬФА-БАНК" г. Новосибирск
Р/с 40702810423010002769 К/с 30101810600000000774 БИК 045004774;
ОГРН 1207000000305 ИНН: 7017469940; КПП 701701001

Исх. №17

06.06.2023г

СПРАВКА О ВНЕДРЕНИИ

Настоящая справка подтверждает то, что результат выпускной квалификационной работы студентки НИ «Томского Государственного Университета» группы 931921 очной формы обучения Овсянниковой Ксении Александровны на тему «Чат-бот для IT-собеседований в Telegram, использующий нейронную сеть» внедрен в рабочий процесс.

Результат выпускной квалификационной работы Овсянниковой Ксении Александровны используется в автоматизации найма IT-специалистов в IT-компанию ООО «Диджитал Номадс».

Директор:



Д.В.Халбаев

СПРАВКА

Томский Государственный Университет

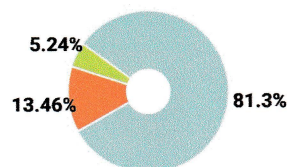
о результатах проверки текстового документа
на наличие заимствований

ПРОВЕРКА ВЫПОЛНЕНА В СИСТЕМЕ АНТИПЛАГИАТ.ВУЗ

Автор работы: Овсянникова Ксения Александровна
Самоцитирование
рассчитано для: Овсянникова Ксения Александровна
Название работы: Овсянникова_ВКР
Тип работы: Выпускная квалификационная работа
Подразделение: НИ ТГУ, Институт прикладной математики и компьютерных наук

РЕЗУЛЬТАТЫ

СОВПАДЕНИЯ	13.46%
ОРИГИНАЛЬНОСТЬ	81.3%
ЦИТИРОВАНИЯ	5.24%
САМОЦИТИРОВАНИЯ	0%



ДАТА ПОСЛЕДНЕЙ ПРОВЕРКИ: 06.06.2023

Структура документа: Проверенные разделы: титульный лист с.1, основная часть с.2, 4-30, содержание с.3, библиография с.31
Модули поиска: ИПС Адилет; Библиография; Сводная коллекция ЭБС; Интернет Плюс*; Сводная коллекция РГБ; Цитирование; Переводные заимствования (RuEn); Переводные заимствования по eLIBRARY.RU (EnRu); Переводные заимствования по коллекции Гарант: аналитика; Переводные заимствования по коллекции Интернет в английском сегменте; Переводные заимствования по Интернету (EnRu); Переводные заимствования по коллекции Интернет в русском сегменте; Переводные заимствования издательства Wiley ; eLIBRARY.RU; СПС ГАРАНТ: аналитика; СПС ГАРАНТ: нормативно-правовая документация; Медицина; Диссертации НББ; Коллекция НБУ; Перефразирования по eLIBRARY.RU; Перефразирования по СПС ГАРАНТ: аналитика; Перефразирования по Интернету; Перефразирования по Интернету (EN); Перефразированные заимствования по коллекции Интернет в английском сегменте; Перефразированные заимствования по коллекции Интернет в русском сегменте; Перефразирования по коллекции

С результатами ознакомлена
- Е.Маш

Работу проверил: Пахомова Елена Григорьевна

ФИО проверяющего

Дата подписи:

06.06.2023

- Е.Маш

Подпись проверяющего



Чтобы убедиться
в подлинности справки, используйте QR-код,
который содержит ссылку на отчет.

Ответ на вопрос, является ли обнаруженное заимствование
корректным, система оставляет на усмотрение проверяющего.
Предоставленная информация не подлежит использованию
в коммерческих целях.