

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический
университет имени К.И.Сатпаева

Институт автоматизации и информационных технологий

Кафедра кибербезопасность, обработка и хранение
информации

Абдуханов Олжас Аскарулы

Разработка и реализация системы «Chatbot» для автоматизации взаимодействия
между преподавателем и студентом в университете

ДИПЛОМНЫЙ ПРОЕКТ

Специальность 5В070300 – Информационные системы

Алматы 2022

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический
университет имени К.И.Сатпаева

Институт автоматизации и информационных технологий

Кафедра кибербезопасность, обработка и хранение
информации



ДОПУЩЕН К ЗАЩИТЕ

Заведующий кафедрой КОиХИ
к.т.н, ассоц. профессор

Р.Ж.Сатыбалдиева

" 19 " 05 20 22 г.

ДИПЛОМНЫЙ ПРОЕКТ

На тему: «Разработка и реализация системы «Chatbot» для
автоматизации взаимодействия между преподавателем и студентом в
университете»

по специальности 5В070300 – Информационные системы

Выполнил:

Абдуханов О.А.

Рецензент

доктор PhD, доцент
доцент

О.В. Киселева
В.А.

" 19 " 05 2022 г

Научный руководитель

канд. технич. наук,

Р.С. Косников

" 19 " 05 2022 г

Алматы 2022

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский технический университет имени
К.И.Сатпаева

Институт автоматизации и информационных технологий

Кафедра кибербезопасность, обработка и хранение информации

5B070300 – Информационные системы



УТВЕРЖДАЮ

Заведующий кафедрой КОиХИ

к.т.н., ассоц. профессор

Р.Ж.Сатыбалдиева

«05» 05 20__ г.

ЗАДАНИЕ

на выполнение дипломного проекта

Обучающемуся: Абдуханову Олжасу Аскарулы

Тема:

Утверждена приказом Ректора Университета №_489-П/Ө от 24.12.2021.

Срок сдачи законченной работы:

Исходные данные к дипломному проекту: описание проекта, документация по среде разработки, техническое задание, описание реализаций проекта.

Краткое содержание дипломной работы:

- а) Анализ предметной области и постановка задач
- б) Инструменты и технологий проектирования
- в) Сбор датасета
- г) Демонстрация работы информационной системы

Перечень графического материала (с точным указанием обязательных чертежей): представлены 15 слайдов презентации.

Рекомендуемая основная литература: из 16 наименований.

ГРАФИК

подготовки дипломной работы (проекта)

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
1. Анализ существующих чат-ботов Постановка цели и задачи.	15.02.2022	
2. Сравнительный анализ среды разработки с другими.	01.03.2022	
3. Проектирование информационной системы Чат-Бот.	25.03.2022	
4. Реализация интерфейса и логической структуры клиент-серверной части.	10.04.2022	

Подписи

консультантов и нормоконтролера на законченную дипломную работу (проект) с указанием относящихся к ним разделов работы (проекта)

Наименования разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Основная часть	Косников В.А. Канд. технич. наук, доцент	10.05.2022 г	
Нормоконтролер	Аристомбаева М.Т. Магистр техн.наук, лектор	16.05.2022 г	

Научный руководитель:  Косников В.А

Задание принял к исполнению обучающийся:  Абдуханов О.А

Дата

"28" января 2022

АННОТАЦИЯ

Цель дипломной работы – разработка чат-бот системы для автоматизации отношений между преподавателем и студентом. Данная система решает проблему простоты интеграции в различные цифровые платформы без необходимости загрузки приложения, сокращает время затрачиваемое на организацию и выполнение задач, поскольку чат-боты дают немедленные ответы, заранее спроектированные, на частые вопросы студентов.

При разработке системы было проведено исследование предметной области, по результатам которого были определены основные задачи и комплекс технологий для их решения. Были применены сервер для передачи данных front и back-end, инструменты веб-парсинга, база данных mongo db и язык программирования node javascript.

При нажатии кнопок бот реагирует на них как на команды. Данный чат-бот подходит для сбора данных, ответов на вопросы и сбор данных. В главном разделе 4 кнопки: «Предметы», «Add/Drop», «Мероприятия», «Написать». Каждая ведет по своему «дереву диалога», предлагая информацию и уточняя запросы.

АНДАТПА

Дипломдық жұмыстың мақсаты – оқытушы мен студент арасындағы қарым-қатынасты автоматтандыру үшін чат-бот жүйесін жасау. Бұл жүйе қосымшаны жүктеп алуды қажет етпей-ақ әртүрлі цифрлық платформаларға интеграциялаудың қарапайымдылығы мәселесін шешеді, тапсырмаларды ұйымдастыруға және орындауға кететін уақытты қысқартады, өйткені чат-боттар студенттердің жиі қойылатын сұрақтарына алдын ала әзірленген жылдам жауаптарды береді.

Жүйені әзірлеу кезінде пәндік саланы зерттеу жүргізілді, оның нәтижелері бойынша негізгі міндеттер мен оларды шешудің технологиялар кешені анықталды. Сервер алдыңғы және артқы жағындағы деректерді, веб-талдау құралдарын, mongo db дерекқорын және түйіннің JavaScript бағдарламалау тілін тасымалдау үшін пайдаланылды.

Түймелерді басу боттың реакциясын бастайды: ол оларға командалар сияқты әрекет етеді. Мұндай бот деректерді жинауға, сұраулар тақырыбын анықтауға, стандартты сұрақтарға жауап беруге жарамды. Негізгі бөлімде 4 түйме бар: «Элементтер», «Қосу/Тастау», «Оқиғалар», «Жазу». Әрқайсысы ақпаратты ұсынып, сұранысты түсіндіре отырып, өзінің «диалог ағашы» арқылы жүргізеді.

THE ANNOTATION

The purpose of the thesis is to develop a chat-bot system for automating the relationship between a teacher and a student. This system solves the problem of ease of integration into various digital platforms without the need to download an application, reduces the time spent on organizing and completing tasks, since chatbots provide immediate answers, pre-designed, to frequent questions from students.

When developing the system, a study of the subject area was carried out, the results of which identified the main tasks and a set of technologies for their solution. A server was used to transfer front and back-end data, web parsing tools, a mongo db database, and the node javascript programming language.

Pressing the buttons initiates the reaction of the bot: it reacts to them as if they were commands. Such a bot is suitable for collecting data, determining the subject of requests, answering standard questions. There are 4 buttons in the main section: "Items", "Add/Drop", "Events", "Write". Each leads through its own "dialogue tree", offering information and clarifying the request.

СОДЕРЖАНИЕ

Введение.....	-10 -
1 Анализ предметной области, постановка задач дипломирования.....	12
1.1 Обзор Чат-Бот Систем.....	12
1.2 Применение чат-бота в образовательной системе.....	13
1.3 Постановка задачи.....	13
2 Инструменты технологии.....	14
2.1 Среда разработки.....	14
2.1.1 WebStorm.....	14
2.1.2 Преимущества и недостатки.....	14
2.1.3 Сравнительный анализ с другими IDE.....	15
2.1.4 Вывод.....	16
2.2 Фронтэнд.....	16
2.2.1 Инструмент разработки.....	16
2.2.2 Преимущества.....	16
2.2.3 Недостатки React JS.....	18
2.2.4 Выводы.....	18
2.3 Бэкэнд.....	18
2.3.1 Общая информация.....	18
2.3.2 Инструмент разработки.....	19
2.3.3 Преимущества nodejs.....	-19-
2.3.4 Недостатки Node.js.....	21
2.3.5 Сравнительный анализ средств разработки.....	21
2.3.6 Выводы.....	22
2.4 Инструменты для сбора информации.....	23
2.5 Модули и библиотеки использованные на языке JavaScript.....	23
3.СБОР ДАТАСЕТА.....	25
3.1 Общая информация.....	25
3.2 Интеграция с существующей системой.....	25

3.3 Загрузка и структурирование запросов	26
3.4 Анализ полученных данных и выбор баз данных	26
3.5 RestApi и Авторизация в системе	27
4. Реализация экспертной системы	30
4.1 Серверная часть	30
4.2 Клиентская часть	31
4.3 Настройка бота	33
4.4 Интерфейс и тестирование чат-бота.....	35
5.Реализация чат-бота в информационной образовательной системе	36
Заключение	38
Список использованной литературы.....	39
Приложение А	39
Приложение Б.....	39

ВВЕДЕНИЕ

В век цифровых технологий возможно создать приложение, которое будет полезно для людей. Одним из таких приложений является Чат-Бот система для автоматизаций отношений между преподавателем и студентом, повышая качество учебного процесса.

Чат-Бот - это умный помощник способный помочь преподавателю сэкономить личное время, которое он тратит на ответы на часто задаваемые вопросы студентов. Для студента – эта система поможет получить моментальные ответы на свои возникшие вопросы, не тревожа при этом преподавателя.

Чат-Бот система не просто онлайн-чат с ответами на вопросы, но также она ускоряет процесс системы Add/Drop предмета, где можно выбрать или удалить предметы самому, не стоя в очереди в офис регистратора.

Чат-Бот помощник всегда подскажет о ближайших мероприятиях, которые будут проходить в стенах университета. Такая система оптимизирует взаимоотношения преподавателя и студента.

ПЕРЕЧЕНЬ ТЕРМИНОВ

Веб-приложение – это непосредственное взаимодействие клиента с сервером.

Отладка - этап разработки, где выявляют недочеты программы и устраняют ошибки.

Интегрированная среда разработки - программных средств, используемая программистами для разработки программного обеспечения.

Архитектура программного обеспечения – это структура из которого была составлена программа.

Брандмауэр – это защитная стена, выполняющая проверку и фильтрацию данных, которые поступают из интернета.

Фреймворк - программная платформа, определяющая структуру программной системы;

Масштабируемость - в электронике и информатике означает способность системы, сети или процесса справляться с увеличением рабочей нагрузки при добавлении ресурсов.

Виртуальная память - метод управления памятью компьютера, при котором задействуется вторичное хранилище при недостатке памяти для выполнения программы.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ, ПОСТАНОВКА ЗАДАЧ

1.1 Обзор Чат-Бот Систем

Развитие технологий чат-ботов было таким же динамичным, как и эволюция возможностей чат-ботов. На данный момент чат-боты можно условно разделить на три типа. Эти три типа различаются по их технической сложности, а именно:

1. Простые чат-боты,
2. Умные чат-боты и
3. Гибридные чат-боты.

1. Простые чат-боты

Простые чат-боты имеют ограниченные возможности и обычно называются ботами, основанными на правилах. Они зависят от конкретной задачи. Думайте о них как о IVR в чате. Это означает, что бот задает вопросы на основе заранее определенных вариантов, и клиент может выбирать из вариантов, пока не получит ответы на свой запрос. Чат-бот не будет делать никаких выводов из своих предыдущих взаимодействий. Эти чат-боты лучше всего подходят для прямых диалогов. Их очень просто построить [9] и обучить. Пример: Заказ пиццы.

Когда клиент взаимодействует с чат-ботом, чтобы заказать пиццу, задается ход разговора. Точно так же, как оператор запрашивает ваш заказ по телефону, чат-бот будет задавать вопросы таким же образом. Начиная от размера пиццы, заканчивая корочкой, начинкой и количеством сыра. Затем он запросит адрес и способ оплаты. Эти шаги логичны и требуют от клиента только перехода по ссылке, чтобы завершить свой заказ.

2. Умные чат-боты

Интеллектуальные чат-боты с поддержкой искусственного интеллекта предназначены для имитации взаимодействия с клиентами, близкого к человеческому. Они могут вести непринужденные беседы и понимать намерения, язык и чувства. Эти чат-боты требуют программирования, чтобы помочь им понять контекст взаимодействий. Их гораздо сложнее реализовать и выполнить, и для их изучения требуется много данных. Пример: Виртуальные помощники.

3. Гибридные чат-боты

Они представляют собой комбинацию простых и умных чат-ботов. Как простые, так и умные чат-боты - это крайности в спектре чат-ботов. Постоянно будет возникать потребность в том, чтобы простые чат-боты были умнее, а умные чат-боты - проще. Гибридные чат-боты отвечают этой золотой середине. У гибридных чат-ботов есть некоторые задачи,

основанные на правилах, и они могут понимать намерения и контекст. Это делает их сбалансированным инструментом взаимодействия бизнеса с клиентами. Пример: Медицинский Диагноз

1.2 Применение чат-бота в образовательной системе

Чат-бот круглосуточно помогает студентам получить ответы на интересующие их вопросы. С помощью бота университет снял с себя нагрузку в период Add/Drop недели.

Что касается взаимодействия, Чат-боты - отличный способ получить обратную связь от учеников или наоборот, предоставить учащимся информацию об их успеваемости.

Чат-бот система может стать правой рукой преподавателя, освобождая его от ряда рутинных задач - например, отвечать, когда нужно сдавать ту или иную работу. Он поможет учителям автоматизировать задачи, тем самым сэкономив время и силы на творческую и важную работу.

Чат-бот не требует создания отдельной платформы - его можно реализовать на базе портала университета sso.satbayev.university. Заполнить бота информацией довольно просто, для этого была разработана админ-панель куда можно заносить, корректировать и удалять данные.

1.3 Постановка задачи

В результате проведенного анализа использования чат-ботов для образовательного процесса в данной дипломной работе были поставлены следующие задачи:

- 1) Оптимизация рабочего времени Преподавателя, путем освобождения от ответов на часто задаваемые вопросы.
- 2) Создание онлайн-ассистента для студента, где он может задать вопрос и получить ответ моментально.
- 3) Информировать студента о предстоящих научных и общественных мероприятиях.
- 4) Оптимизировать систему Add/Drop предмета для скоростного достижения студентом.

2 ИНСТРУМЕНТЫ И ТЕХНОЛОГИИ

2.1 Среда разработки

2.1.1 WebStorm

WebStorm - это интуитивно понятная, платная и лицензированная среда разработки JavaScript, созданная JetBrains для интерфейсной разработки и серверного JavaScript. Основные характеристики WebStorm: это очень всеобъемлющая и интеллектуальная среда разработки для легкого запуска, отладки и модульного тестирования Node.js приложения, а также имеет хороший рефакторинг кода и автоматический импорт. В отличие от этого, легкая, без излишеств WebStorm IDE [6] основана на платформе IntelliJ от JetBrains. Он предоставляет функции, направленные на упрощение и ускорение процесса разработки, а также плагины CSS и JavaScript, идеально подходящие для небольших проектов веб-разработки.

Также, он имеет впечатляющий набор функций, которые делают его конкурентоспособной IDE, особенно для веб-приложений на основе JavaScript. Другие поддерживаемые языки, форматы и фреймворки включают TypeScript, Markdown, JSON, YAML, React, AngularJS и Node.js .

2.1.2 Преимущества и недостатки

Каковы плюсы и минусы WebStorm, IDE JetBrains для JavaScript?

Прежде всего, у WebStorm есть бесплатная 30-дневная пробная версия, так что вы можете попробовать ее бесплатно и самостоятельно определить ее плюсы и минусы.

Готовый опыт. Можно сразу приступить к кодированию без необходимости установки каких-либо расширений. Поддержка IDE всех основных фреймворков и технологий, таких как Angular, Vue, React, webpack и Git, будет просто работать.

Возможности рефакторинга. В WebStorm есть довольно много рефакторингов кода, и все они работают надежно.

Хорошо работает с большими кодовыми базами. Все ключевые функции, от рефакторинга кода до навигации, хорошо работают даже с большими кодовыми базами.

Расширенная интеграция VCS. Большая часть вашей работы с Git, GitHub и другими VCS может быть выполнена с помощью графического интерфейса WebStorm. Для этого нет необходимости использовать сторонний клиент или эмулятор терминала.

Позволяет выполнять большую часть работы в одном месте. В дополнение к поддержке VCS, WebStorm имеет встроенный отладчик, эмулятор терминала, интеграцию с инструментами тестирования и многое другое, что означает, что вы можете выполнять множество задач, не покидая его.

Основные недостатки:

Медленнее, чем текстовые редакторы. Будучи IDE с множеством предопределенных функций, WebStorm действительно не так быстр, как текстовые редакторы, такие как Sublime Text или VS Code. Но если привыкнуть к WebStorm и начнете активно использовать его функции, это значительно повысит производительность.

К этому трудно привыкнуть. Поскольку существует довольно много функций, кнопок и ярлыков, может быть трудно привыкнуть к WebStorm, особенно если никогда раньше не использовали IDE. Хорошей новостью является то, что существует довольно много образовательных ресурсов, которые могут вам помочь. Например, есть плагин Key Promoter X, который может помочь вам освоить ярлыки, а также руководство WebStorm с полезными советами и рекомендациями для получения большего от IDE.

2.1.3 Сравнительный анализ с другими IDE

Интегрированная среда разработки предоставляет инструменты и функции программирования, которые превосходят действия при проектировании и облегчают разработчикам переписывание или рефакторинг кода. Visual Studio от Microsoft и WebStorm от JetBrains, возможно, представляют собой две разные стороны рынка интегрированных сред разработки, когда речь заходит о возможностях разработки, поддержке определенных языков и, конечно же, стоимости.

Что такое Visual Studio?

Microsoft Visual Studio - это интегрированная среда разработки (IDE), которая поддерживает широкий спектр хорошо зарекомендовавших себя языков разработки и форматов данных, таких как C, C++, JavaScript, XML, XSLT, HTML и CSS. В то время как основа Visual Studio (что неудивительно) основана на платформе Microsoft .NET Framework, она может обрабатывать целый ряд подходов к разработке программного обеспечения, и все они поддерживаются собственной поддержкой сообществ разработчиков Microsoft.

Кроме того, расширение ReSharper от Visual Studio обеспечивает анализ кода перед компиляцией, который может обнаруживать и выделять ошибки прямо в редакторе. Visual Studio также предоставляет ряд шаблонов и инструментов для тестирования, включая IntelliTest для модульного тестирования в реальном времени и тестов, управляемых пользовательским интерфейсом.

Как и Visual Studio, WebStorm может похвастаться набором функций динамического и статического анализа кода, которые помогают обнаруживать языковые ошибки и ошибки во время выполнения. Разработчики могут расширить этот анализ, чтобы искать такие вещи, как неиспользуемые переменные и несуществующие методы. И Visual Studio, и WebStorm хорошо разбираются в отладке Node.js и приложения React, и оба предлагают поддержку приложений на основе JavaScript. Однако

принадлежность WebStorm к JetBrains дает ему небольшое преимущество, когда дело доходит до рефакторинга кода JavaScript и TypeScript, особенно в тех случаях, когда функции и переменные должны передаваться между файлами.

2.1.4 Вывод

WebStorm - это мощная среда разработки на JavaScript, которая помогает быстро писать высококачественный код, независимо от того, насколько сложными могут быть ваши проекты. Благодаря встроенной поддержке всех популярных технологий и множеству встроенных инструментов разработчика он позволяет вам сразу приступить к работе, вместо того чтобы тратить время на настройку среды разработки.

2.2 Фронтэнд

2.2.1 Инструмент разработки

Интерфейс чат-бота разработан с использованием React.js [2] - интерфейсная библиотека JavaScript с открытым исходным кодом для создания пользовательских интерфейсов на основе компонентов пользовательского интерфейса. Кроме того, React.js помогает в разработке адаптивных веб-страниц.

Многофункциональная библиотека JavaScript где открытый исходный код. Он способен отображать интерфейсы веб-приложений, которые являются простыми, быстрыми и масштабируемыми.

Опрос, проведенный Stack Overflow, показал, что React на самом деле является самым популярным фреймворком для разработки интерфейсов, занимая 1-е место. Более 220 000 веб-сайтов сегодня используют React; Apple, BBC, PayPal и Netflix являются одними из международных гигантов, которые используют эту платформу для своих приложений.

2.2.2 Преимущества

Прежде чем мы рассмотрим, почему он так популярен, нужно должным образом ознакомимся с функциями React JS. Это поможет понять, как работает React JS.

1. Виртуальный DOM

Эта характеристика React помогает ускорить процесс разработки приложений и обеспечивает гибкость. Алгоритм облегчает репликацию веб-страницы в виртуальной памяти React. Всякий раз, когда приложение модифицируется или обновляется, весь пользовательский интерфейс снова отображается виртуальным DOM путем обновления компонентов, которые были изменены. Это сокращает время и затраты, затрачиваемые на разработку.

Таким образом, разработчики могут гарантировать превосходный пользовательский интерфейс и высокую производительность приложения.

2. JavaScript XML или JSX

Это синтаксис разметки, который описывает внешний вид интерфейса приложения. Он делает синтаксис таким же, как у HTML, и используется

разработчиками для создания компонентов React.JSX - одна из лучших функций React JS, поскольку она очень упрощает для разработчиков написание строительных блоков.

3. React Native

Использует собственные, а не веб-компоненты для облегчения разработки собственного React JS для Android и iOS.

Эта функция преобразует код React, чтобы сделать его совместимым с платформами iOS или Android, и предоставляет доступ к их встроенным функциям.

4. Декларативный пользовательский интерфейс

Эта функция делает код React более читаемым и облегчает исправление ошибок. React JS - лучшая платформа для разработки увлекательных и привлекательных пользовательских интерфейсов не только для веб-приложений, но и для мобильных приложений.

5. Архитектура На основе Компонентов

Это просто означает, что пользовательский интерфейс приложения, основанного на React JS, состоит из нескольких компонентов, каждый из которых имеет свою особую логику, написанную на JS.

Благодаря этому разработчики могут передавать данные по всему приложению без ущерба для DOM. Компоненты React JS играют огромную роль в определении визуальных эффектов и взаимодействий приложения.

6. Короткая Кривая Обучения

По сравнению с другими фреймворками, React JS очень прост в освоении, что поднимает его в чартах популярности; это позволяет разработчикам завершать свои проекты в рекордно короткие сроки.

7. Это Повышает Производительность Разработчиков

Благодаря функции повторного использования компонентов, позволяет разработчикам распространять одни и те же цифровые объекты.

Поэтому даже при частых обновлениях или модификациях компонентов, которые обычно влияют на другие, в React это не становится проблемой для разработчиков.

Разработчики могут просто добавлять текстовые поля, кнопки или другие простые элементы, а на следующем этапе размещать их в компонентах-оболочках и, в конечном итоге, в корневом элементе.

Преимущество здесь заключается в том, что он обеспечивает улучшенное обслуживание и расширение кода, поскольку каждый компонент React имеет уникальную внутреннюю логику, которой можно легко управлять.

8. Полезный набор инструментов для разработчиков

Facebook включил несколько инструментов разработчика для React и Chrome в платформу react js, и они очень помогают разработчикам в обнаружении родительских и дочерних компонентов, наблюдении за их иерархией, изучении их текущего состояния и так далее.

Благодаря самообучению разработчики смогут использовать новые технологии в реальных, реальных проектах.

9. Повышенная стабильность Кода

Нисходящий поток данных в React Js гарантирует, что изменения, внесенные в дочернюю структуру, не повлияют на родительскую структуру.

Чтобы изменить объект, разработчику нужно только отредактировать его состояния и внести соответствующие изменения; таким образом, будет обновлен только определенный компонент, в то время как остальные останутся неизменными. Следовательно, улучшается стабильность кода, а производительность приложения также очень плавная.

2.2.3 Недостатки React JS

Как и в случае с любой платформой или фреймворком, несмотря на свои многочисленные преимущества, React JS имеет некоторые недостатки.

К счастью, их очень мало. Они являются:

- a. Отсутствие адекватной документации – несмотря на наличие большого сообщества пользователей и разработчиков React, существует потребность в дополнительной документации.
- b. Охватывает только пользовательский интерфейс – невозможно разработать все приложение, используя только react js; нам также нужно использовать другие языки.
- c. Высокие темпы – разработка приложений на основе React происходит очень быстро, и это может оказать большое давление на разработчика.
- d. Перенос данных из React Js может занять много времени и быть громоздким; если вы не уверены в будущем приложения, вам, возможно, придется пойти в другом направлении для вашего frontend framework.
- e. Настройка React может занять больше времени, чем у других популярных фреймворков, таких как Vue.

2.2.4 Выводы

Учитывая плюсы и минусы Reactjs, его можно легко описать тремя словами: безрисковый, отзывчивый и продвинутый. Основная идея, лежащая в основе этой конкретной библиотеки, заключается в следующем: “создавать крупномасштабные приложения с данными, которые постоянно меняются с течением времени”, и она хорошо справляется с этой задачей. Он предоставляет разработчикам возможность работать с виртуальным браузером (DOM), который намного быстрее и удобнее в использовании, чем реальный.

2.3 Бэкэнд

2.3.1 Общая информация

Чат-боты извлекают информацию, необходимую для выполнения требуемых задач, из Серверной части, а затем пересылают сообщение Компоненту управления диалогами и генерации ответов

Чат-ботам, основанным на правилах, требуется База данных для хранения созданных вручную правил. Правила в Базе данных должны быть как можно более разнообразными и целостными, чтобы обеспечить надежность чат-бота.

Разработка БД необходима и обязательна, но может быть очень трудоемкой и требовательной, поскольку она включает в себя ручную работу.

2.3.2 Инструмент разработки

В качестве среды разработки серверной части был выбран node.js.

Выбор node.js обусловлен тем, что node.js превосходная и легко адаптируемая среда со многими подходящими функциональными возможностями.

Более того, он работает в однопоточном цикле, основанном на событиях, чтобы сделать все выполнения неблокирующими. Однако вы не можете рассматривать Node.js поскольку это не фреймворк, и он не имеет ничего общего с PHP, .NET или JAVA.\

2.3.3 Преимущества nodejs

1. Node.js обеспечивает легкую масштабируемость

Одним из ключевых преимуществ Node.js заключается в том, что разработчикам легко масштабировать приложения как в горизонтальном, так и в вертикальном направлениях. Приложения могут быть масштабированы горизонтально путем добавления дополнительных узлов к существующей системе.

Более того, Node.js также предлагает вам возможность добавления дополнительных ресурсов к отдельным узлам во время вертикального масштабирования приложения. Таким образом, он обладает высокой масштабируемостью и предоставляет лучшие возможности, чем другие серверы JavaScript.

2. Преимущества Fullstack JS

Node.js рассматривался как полнофункциональный JavaScript для обслуживания как клиентских, так и серверных приложений.

Таким образом, преимущество заключается в том, что вам не нужно нанимать отдельных разработчиков как для серверной, так и для интерфейсной разработки. Это экономит как ваши ценные деньги, так и время.

3. Известен тем, что предлагает высокую производительность

Ранее уже упоминалось, что Node.js интерпретирует код JavaScript с помощью движка JavaScript V8 от Google. Этот движок преобразует код JavaScript непосредственно в машинный код. Это упрощает и ускоряет эффективную реализацию кода. Скорость выполнения кода также повышается за счет среды выполнения, поскольку она поддерживает неблокирующие операции ввода-вывода.

4. Преимущество кэширования

Среда выполнения с открытым исходным кодом для Node.js также предоставляет возможность кэширования отдельных модулей. Разработчикам не нужно повторно выполнять коды, поскольку кэширование позволяет приложениям быстрее загружать веб-страницы и быстрее реагировать на действия пользователя.

5. Предоставляет свободу в разработке приложений

Еще одно преимущество, которое Node.js разработчикам предлагается свобода разработки приложений и программного обеспечения.

Это одна из важных функций, которая по-прежнему отсутствует в Ruby on Rails, налагая определенные правила. Вы можете начать все с нуля при разработке приложений.

6. Получение поддержки для часто используемых инструментов

С Node.js , разработчики могут получить расширенную поддержку различных часто используемых инструментов. Давайте возьмем пример. Предположим, вы хотите протестировать исходный код Node.js приложение; вы можете сделать это с помощью Jasmine и других подобных инструментов модульного тестирования.

Аналогичным образом, если вы хотите определить и установить зависимости проекта, вы можете использовать npm, мощный менеджер пакетов. Вы можете использовать grunt для выполнения задач проекта.

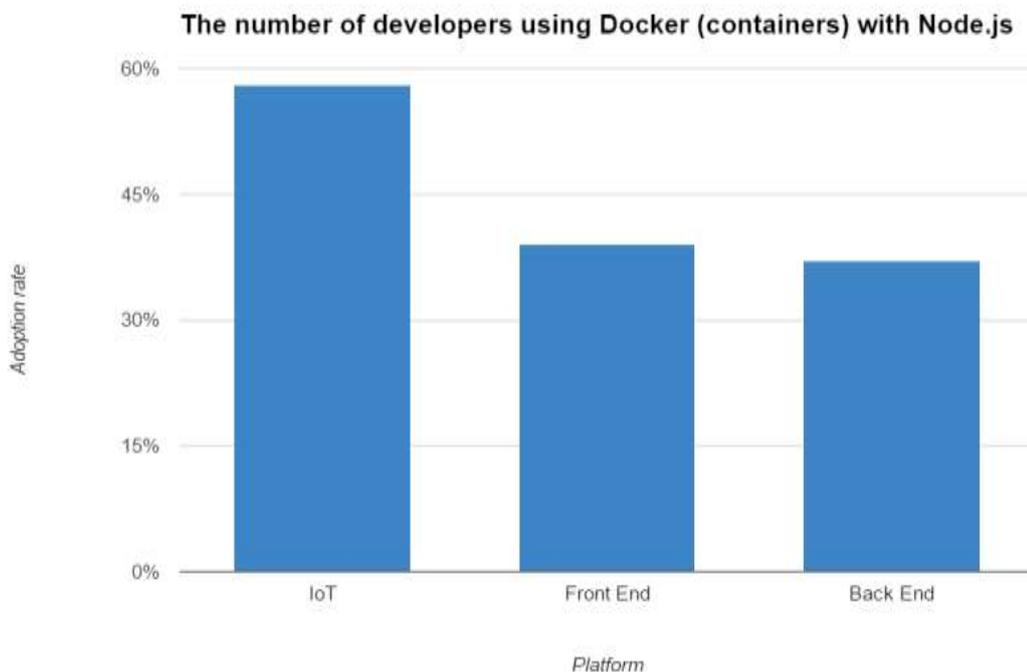


Рисунок 1 – График использования node.js

2.3.4 Недостатки Node.js

Node.js также имеет некоторые минусы, которые необходимо обязательно обсудить.

1. Интерфейс прикладного программирования (API) нестабилен

Одно из главных недочетов, с которой сталкиваются это частое изменение интервалов в интерфейсе прикладного программирования (API).

Время от времени появляется новый API с рядом изменений, несовместимых с обратной связью. В результате разработчики вынуждены вносить изменения в доступные базы кода, чтобы обеспечить совместимость с последней версией Node.js API.

2. Снижает производительность при выполнении сложных вычислительных задач

Несмотря на то, что он является однопоточным и управляемым событиями, Node.js не справляется с выполнением тяжелых вычислений на базе ЦП по той же причине. Когда Node.js получает значительную задачу, управляемую процессором, в своем цикле событий, он использует всю доступную мощность процессора для выполнения этой задачи. Это приводит к замедлению общего цикла событий, что еще больше затрудняет работу интерфейса вашего приложения.

Чтобы решить эту проблему, в 2018 году Node.js введен модуль “рабочие потоки”, который позволял разработчикам выполнять несколько потоков одновременно. Однако это решение не полностью позволяет справляться с вычислительными задачами, связанными с ЦП.

3. Отсутствие поддержки библиотеки может поставить под угрозу ваш код.

Существует несколько реестров и библиотек npm, которые либо некачественны, либо неполны и не задокументированы надлежащим образом. Следовательно, этот неадекватный мониторинг затруднен, если некоторые любители разрабатывают веб-приложение в Node.js .

2.3.5 Сравнительный анализ средств разработки

Node.js против Ruby on Rails против Django против Symfony

Существует не так много технологий, которые могут конкурировать с Node.js - популярность или рыночный спрос. Однако, если посмотреть на альтернативы для серверного программирования, мы увидим по крайней мере еще три среды разработки. Это - Ruby on Rails, Django и Symfony.

Rails не может конкурировать с производительностью и масштабируемостью Node, однако Rails может быть лучшим выбором для быстрой разработки. Node.js действительно, для достижения такой скорости разработки приходится использовать сторонние модули, но у Ruby все это нетрадиционно.

Django - это веб-фреймворк на Python, самом быстрорастущем языке программирования, согласно данным StackOverflow [5]. Считается, что Django, наряду с Python, также имеет более мягкую кривую обучения. Во время работы с Node.js потребуются обширные знания JavaScript. По сравнению с Node.js, Django - это удобный инструмент для начинающих. Кроме самого языка, нет никаких реальных причин выбирать один из них вместо другого. Это скорее вопрос вашего комфорта и опыта работы с JavaScript или Python, чем с конкретными вариантами использования.

Symfony - это фреймворк PHP, который автоматически предоставляет ему более чем 20-летнюю документацию и огромное активное сообщество. Хотя Symfony - всего лишь один из фреймворков PHP на рынке, он впечатляюще стабилен, масштабируем и хорошо работает для крупномасштабных проектов. Symfony, естественно, поддерживает функции CMSs, такие как шаблоны и панели администрирования, что позволяет ему запускать блоги, новостные сайты и магазины электронной коммерции. Некоторые из примеров включают Yahoo Answers, Dailymotion и National Geographic.

Очевидно, что со всеми перечисленными Node.js преимуществами и недостатками, технология не является серебряной пулей. Но ни один из них не является Java, .Net framework или PHP. Тем не менее, существуют конкретные случаи, когда каждая из перечисленных технологий работает наилучшим образом. Для Node.js, это приложения реального времени с интенсивным вводом-выводом, требующие скорости и масштабируемости.

Это могут быть социальные сети, игровые приложения, живые чаты или форумы, а также программное обеспечение для фондовых бирж или рекламные серверы, где скорость - это все. Быстрый и масштабируемый, Node.js это технология выбора для устройств и приложений Интернета вещей с интенсивным использованием данных в режиме реального времени.

Node.js сделал JavaScript предпочтительной технологией полного стека для разработки веб-приложений. Благодаря своей неблокирующей архитектуре, Node.js хорошо работает для кодирования и трансляции видео и аудио, загрузки нескольких файлов и потоковой передачи данных. Последнее может быть чрезвычайно полезно для программного обеспечения туристической индустрии, где вам необходимо получать данные из разных API разных поставщиков.

2.3.6 Выводы

Создание стабильной и надежной серверной части - одна из самых важных вещей в разработке приложений. Вот почему очень важно выбрать правильную технологию, которая будет отвечать всем потребностям. В Node.js является более выгодным для разработчиков по сравнению с его недостатками. Что еще более важно, так это тот факт, что он расширил область применения JavaScript и, очевидно, может использоваться как для интерфейсных, так и для внутренних серверов.

2.4 Инструменты для сбора информации

Администраторам чат-бота часто приходится обновлять чат-бота текущими информацией о мероприятиях, студентах, расписаниях. Это позволит чат-боту предоставлять своим пользователям актуальную и правильную информацию. Кроме того, будет добавлен список студентов с их идентификационными номерами и номером телефона для идентификации личности. Страница была разработана с использованием react.js для интерфейса и node.js для серверной части базы данных.

2.5 Модули и библиотеки использованные на языке JavaScript Mongoose

Модуль Mongoose [14] - один из самых мощных внешних модулей NodeJS. Mongoose - это MongoDB ODM, т.е. (Моделирование объектной базы данных), который использовался для перевода кода и его представления из MongoDB в Node.js сервер.

Преимущества модуля Mongoose :

- Проверка коллекции базы данных MongoDB [3] выполняется легко.
- В коллекции может быть реализована предопределенная структура.
- Ограничения могут быть применены к документам коллекций с помощью Mongoose.
- Модуль Mongoose построен поверх драйвера MongoDB и обеспечивает легкую абстракцию запроса и определение запроса.
- Многие разработчики, которые привыкли использовать SQL, чувствуют себя некомфортно при работе с MongoDB, потому что базы данных Nosql и гибкая структура здесь Mongoose играют важную роль и делают схему сбора данных похожей на базы данных SQL.

Модуль Mongoose предоставляет несколько функций для управления документами коллекции базы данных MongoDB. Реализация для определенной структуры и проверки коллекции: Модуль mongoose наложил определенную структуру на коллекцию и делает коллекцию жесткой.

Установка Модуля:

```
npm install mongoose
```

Express

"Express [13] - это быстрый, непредвзятый минималистичный веб-фреймворк для Node.js [4] "Express.js представляет собой фреймворк веб-приложения для Node.js . Он предоставляет различные функции, которые делают разработку веб-приложений быстрой и простой, что в противном случае занимает больше времени, используя только Node.js [4].

Express.js основан на Node.js модуль промежуточного программного обеспечения называется connect, который, в свою очередь, использует http-

модуль. Таким образом, любое промежуточное программное обеспечение, основанное на connect, также будет работать с Express.js .

Преимущества Express.js:

- Делает Node.js разработка веб-приложений быстрая и простая.
- Легко настраивается и настраивается.
- Позволяет вам определять маршруты вашего приложения на основе методов HTTP и URL-адресов.
- Включает в себя различные модули промежуточного программного обеспечения, которые вы можете использовать для выполнения дополнительных задач по запросу и ответу.
- Легко интегрируется с различными движками шаблонов, такими как Jade, Vash, EJS и т.д.
- Позволяет определить промежуточное программное обеспечение для обработки ошибок.
- Легко обслуживать статические файлы и ресурсы вашего приложения.
- Позволяет создать сервер REST API. [1]rea
- Простое подключение к базам данных, таким как MongoDB, Redis,

MySQL

- Установить Express.js
- Вы можете установить express.js использование npm. Следующая команда установит последнюю версию express.js глобально на вашем компьютере, чтобы каждый Node.js приложение на вашем компьютере может использовать его.

Установка Express

```
npm -g express
```

Axios

Преимущества:

- Выполняет XMLHttpRequests из браузера
- Выполняет http-запросы из node.js
- Поддерживает Promise API
- Перехват запроса и ответа
- Преобразование данных запроса и ответа
- Отмена запросов
- Автоматическое преобразование данных в формате JSON
- Поддержка на стороне клиента для защиты от XSRF

Установка Axios [15]

```
npm install axios
```

3.СБОР ДАТАСЕТА

3.1 Общая информация

При создании чат-бота возникает необходимость сбора информации.

У университета есть много “данных” в виде ответов, которые они хотят предоставить в чат-боте. Однако им часто не хватает самых важных данных, необходимых для обучения чат-бота: вариативность запросов пользователей по интересующим им вопросам, примеров того, как пользователи выражают свои цели и потребности (намерения). Обусловлено это тем что, каждый пользователь имеет своё представление и поведение при задавании вопросов. Например, варианты названия предмета «Физическая культура» : физра, физкультура, фз , физическая культура и т.д.

Чат-бот работает с использованием входных данных, затем они настраиваются для предоставления выходных данных (ответов). Таким образом, данные должны состоять из примеров пользователей, задающих вопросы или делающих запросы. Как только модель обучена, она будет в состоянии классифицировать намерение запроса, даже если формулировка не совсем похожа на примеры, которые она уже видела. Она будет идентифицировать класс запросов по ключевым словам и тэгам, а далее по подготовленным данным, выдавать нужную информацию.

Для того, чтобы собрать данные, было решено запустить бета-версию чат-бота и собрать информацию. По результатам которых стали известны примеры запросов пользователей и их представление запросов.

3.2 Интеграция с существующей системой

Внедрение чат-ботов в практику работы со студентами позволит улучшить взаимодействие с обучающимися, даст им возможность получать ответы на интересующие вопросы, не обращаясь к преподавателю, тем самым избавит преподавателей от необходимости многократно отвечать на одни и те же стандартные вопросы студентов, зачастую уточняющего характера.

Создание чат-ботов требует тщательного структурирования информации, разделения ее на логические блоки, такие как:

1. Данные о предметах
2. Add/Drop
3. Мероприятия
4. Написать

С помощью чат-ботов до пользователей оперативно и в полном объеме доводится информация методического характера:

– информация о кафедре, адрес страницы кафедры на сайте университета, адреса электронной почты преподавателей, график консультаций;

– даты организационных мероприятий, выдержки из приказов факультета, сроки сдачи и защиты курсовых, экзаменов;

- расписание предметов;
- ссылки на электронную библиотечную систему университета с рекомендациями по оформлению библиографического списка:
- бланки документов, которые необходимы за время учёбы;
- контакты преподавателей, эдвайзеров и учебной части;

3.3 Загрузка и структурирование запросов

В начале дипломной работы был задан вопрос, насколько будет актуален разработанный чат-бот и будет ли он востребован в виде веб-страницы. Для этого был проведен онлайн-опрос среди преподавателей, где на вопрос: "Будет ли полезен виртуальный чат-бот при решении рутинных задач?" из 21 ответов преподавателей было получено 17 ответов «да», что было принято, как минимальное обоснование в актуальности данного проекта и последующих гипотез по его развитию.

Ссылка на результаты опроса -

https://docs.google.com/forms/d/e/1FAIpQLSfuo2dSTuYFHkAFgy17tuXvE-4RheU9pj_0TXKpCGHdaHivWw/viewform?usp=sf_link

Далее были заданы вопросы, чтобы узнать интересующие и актуальные для пользователей темы и вопросы, в результате которой были собраны данные впоследствии использующиеся при создании корректного ответа по заданному запросу.

3.4 Анализ полученных данных и выбор баз данных

Все данные включая: расписание предметов и экзаменов, Силлабус, Индивидуальный учебный план, Транскрипт, Журнал, Общее расписание, УМКД, Личные данные, Документы/Инструкции, Задолженности, Контакты преподавателей и эдвайзеров, Справки, Научная библиотека согласовавшись были взяты из официальных источников учебного заведения и достоверных первоисточников. Во избежание ошибок и недоразумений все данные были откорректированы и добавлены в Базу Данных чат-бота через панель администратора. В данной системе используется База Данных MongoDB – где мы можем корректировать, обновлять и удалять данные.

MongoDB - это кроссплатформенная программа баз данных, ориентированная на документы, с открытым исходным кодом.

Модель документов MongoDB позволяет легко моделировать и манипулировать практически любой структурой данных. MongoDB поддерживает создание явных схем и проверку данных. Такая гибкость является невероятным преимуществом при работе с реальными данными и изменениями в требованиях или среде. MongoDB обеспечивают значительное повышение производительности по сравнению с пакетной записью в традиционных базах данных.

Mongo DB и NoSQL

Базы данных NoSQL хранят данные иначе, чем реляционные базы данных; вместо того, чтобы хранить данные в таблицах строк и столбцов, все

записи в базах данных MongoDB представляют собой документы, определенные в двоичном представлении данных, называемом BSON. Эта информация извлекается приложениями в формате JSON [7].

NoSQL означает 'Не только SQL', и в нем есть много типов, таких как столбец, документ, график, пары ключ-значение и т. д.; MongoDB относится к типу документа, как упоминалось выше. В реляционных базах данных разработчикам необходимо переводить таблицы в объектную модель, чтобы сделать их пригодными для использования в приложении; однако теперь в MongoDB NoSQL и хранимые данные, и объектная модель имеют одинаковую структуру BSON.

3.5 RestApi и Авторизация в системе

API (интерфейс прикладной программы) - это набор правил, который позволяет различным программам взаимодействовать друг с другом. В нем описывается подходящий для разработчика программного обеспечения способ создания программы на сервере, который взаимодействует с различными клиентскими приложениями.

Интеграция API относится к паре приложений (двух или более), связанных между собой через свои API для обмена данными и выполнения совместной функции, таким образом, обеспечивая взаимодействие между приложениями. Теперь, когда мы определили API, давайте перейдем к REST API. Различные веб-сайты, такие как Amazon, Google, Facebook, LinkedIn и Twitter, используют API-интерфейсы на основе REST, которые позволяют пользователям взаимодействовать с этими облачными сервисами.

Полная форма REST API - это интерфейс прикладного программирования с передачей репрезентативного состояния, более известный как веб-сервис REST API. Это означает, что при вызове RESTful API сервер передаст представление состояния запрошенного ресурса в клиентскую систему.

REST API использует методы HTTP, описанные протоколом RFC 2616. Он использует следующие HTTP-запросы:

- получить запрос на извлечение данных
- поместить запрос на изменение состояния данных (например, объекта, файла или блока)
- отправить запрос на создание данных
- удалить запрос, чтобы устранить его

Для Чего используются RESTful API?

Чтобы лучше понять определение REST API и то, как он работает, рассмотрим пример.

Вы что-то ищете, и возвращается список результатов от запрошенной вами услуги. В технологии REST предполагается, что все вызовы не имеют состояния. Это означает, что служба REST не может ничего сохранять между выполнением, что делает ее полезной в облачных приложениях. Это связано с тем, что компоненты без состояния могут быть легко переназначены в

случае сбоя. Кроме того, их можно легко масштабировать с учетом изменений нагрузки. Это потому, что любой запрос может быть отправлен любому экземпляру компонента.

Поскольку никакие данные не могут быть сохранены, которые должны быть отозваны последующей транзакцией, REST является желаемым протоколом для интернет-связи. Как указывалось ранее, технология REST API также полезна при подключении к облачным приложениям, поскольку доступ к сервису через API требует корректировки интерпретации URL-адреса.

Аналогичным образом, REST API также можно использовать для сопоставления данных с облачной платформы в хранилище данных или наоборот.

Клиент

Клиент - это аппаратное или программное обеспечение, которое использует API, доступный с помощью сервера. Например, когда вы посещаете веб-сайт Facebook, ваш браузер является клиентом, который вызывает API Facebook и использует отправленные обратно данные для отображения информации на вашем экране.

Ресурс

Ресурсом может быть любой объект, о котором API может предоставить информацию. Например, в случае API Twitter ресурсом может быть пользователь, хэштег или любой тип носителя, например картинка. Каждый ресурс имеет отдельный идентификатор, который может быть именем или номером.

Ресурс является основной абстракцией информации в REST. REST API использует идентификатор ресурса для распознавания конкретного ресурса, участвующего в обмене данными между различными элементами.

Сервер - это любая система, содержащая ресурсы, которые нужны клиенту. Когда он получает запросы клиента, он предоставляет содержимое клиенту с помощью интерфейса API. Сервер предоставит только репрезентативное состояние источника, а не полный доступ к клиенту.

Отличным примером этого является случай, когда мобильное приложение показывает видео с YouTube через свой интерфейс. Он использует REST API для вызова видеоконтента с YouTube без его размещения в своей системе.

Функции REST API

Вот список функций, которые делают его наиболее эффективной интеграцией данных и приложений.

Масштабируемость

REST API обеспечивает отличную масштабируемость. Поскольку клиенты и серверы разделены, команда разработчиков может масштабировать продукт без особых проблем.

Кроме того, проще интегрировать REST с существующими сайтами без рефакторинга инфраструктуры веб-сайта. Это позволяет разработчикам работать быстрее вместо того, чтобы тратить время на переработку веб-сайта с нуля. В качестве альтернативы они могут просто добавить дополнительную функциональность. Это делает его наиболее часто используемым методом интеграции.

Гибкость и мобильность

Пользователи могут легко взаимодействовать, даже если клиент-сервер REST размещен на разных серверах, что является существенным преимуществом с точки зрения руководства.

Независимость

Благодаря разделению между клиентом и сервером протокол REST упрощает автономное выполнение разработок в различных областях. Кроме того, REST API настраивается в соответствии с операционным синтаксисом и платформой, что дает возможность тестировать множество сред во время разработки.

Клиент-Сервер

Этот принцип REST основан на концепции, согласно которой клиент и сервер должны быть изолированы друг от друга и иметь возможность развиваться независимо. Таким образом, вы можете улучшить управляемость на многих платформах и повысить масштабируемость за счет оптимизации серверных компонентов, поскольку проблемы пользовательского интерфейса отделены от проблем хранения данных.

Единый интерфейс

Чтобы отделить клиент от сервера, вам необходимо иметь унифицированный интерфейс, который позволяет автономно разрабатывать приложение без жесткой привязки его сервисов, моделей и действий к самому уровню API.

Многоуровневая Система

Архитектура REST API включает в себя несколько уровней, которые работают вместе, создавая иерархию, которая помогает создавать более масштабируемое и гибкое приложение. Благодаря многоуровневой системе приложение обладает большей безопасностью, поскольку компоненты на каждом уровне не могут взаимодействовать за пределами последующего уровня. Кроме того, он балансирует нагрузку и предлагает общие кэши для стимулирования масштабируемости.

Многоуровневая система архитектуры REST API обладает большей стабильностью, поскольку она ограничивает производительность компонентов. так что каждый компонент не может "видеть" дальше, чем непосредственный слой, с которым он взаимодействует.

4. РЕАЛИЗАЦИЯ ЭКСПЕРТНОЙ СИСТЕМЫ

4.1 Серверная часть

Для серверной части проекта были поставлены нефункциональные требования для оптимальной работы. Нефункциональные требования включают следующее:

(1) Безопасность: неавторизованные пользователи не должны иметь доступа к системе

(2) Удобство использования: предлагаемая система должна быть простой для пользователя в эксплуатации, вводе данных и интерпретации выходных данных

(3) Масштабируемость: система должна работать адекватно в любое время, независимо от обновлений

(4) Совместимость: предлагаемая система должна быть совместима со всеми веб-браузерами

С этими требованиями хорошо справляется Nginx [16]. Nginx, произносится как “engine-eh”, представляет собой веб-сервер где открыт исходный код, а также применяется в качестве обратного прокси.

Первоначально Nginx задумывалась как программное обеспечение как ответ на проблему C10k, которая представляет собой проблему, связанную с производительностью обработки 10 000 одновременных подключений. (см. Рис.2 ниже)

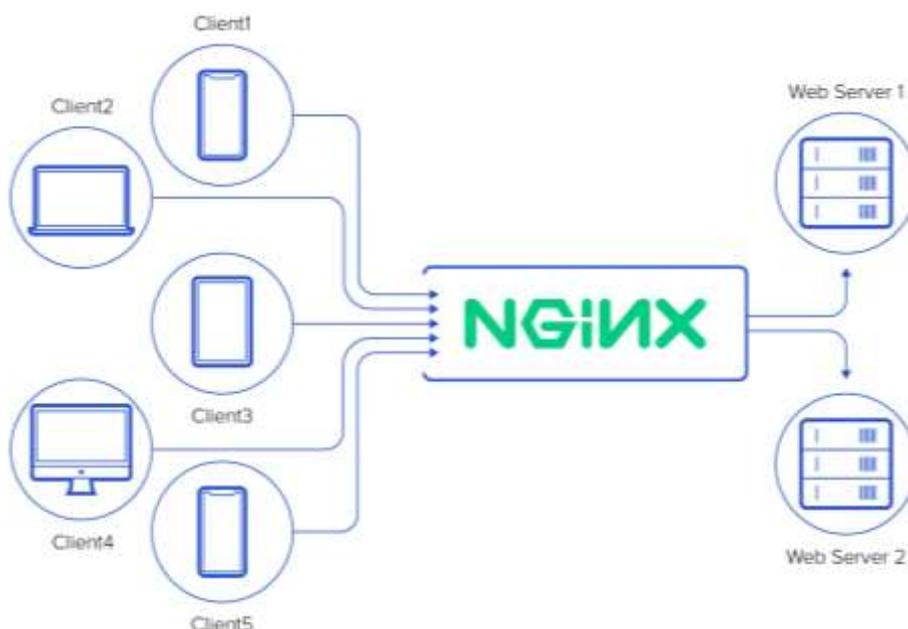


Рисунок 2 – Диаграмма работы Nginx

Как Работает Nginx?

Nginx построен таким образом, чтобы обеспечить низкое использование памяти и высокий уровень параллелизма. Вместо того, чтобы создавать новые процессы для каждого веб-запроса, Nginx использует асинхронный,

управляемый событиями подход, при котором запросы обрабатываются в одном потоке.

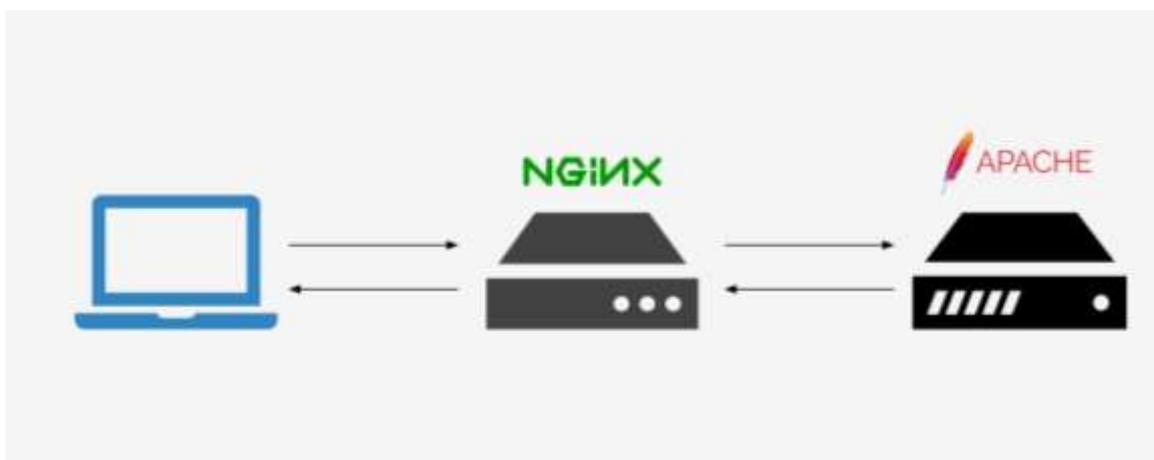


Рисунок 3 – Интеграция Nginx и Apache

Преимущества использования NGINX:

- Написанная кодовая база более последовательна, чем другие альтернативы.
- Он обеспечивает удобный формат конфигурации и имеет современный дизайн, чем любые другие альтернативы веб-серверу.
- Он основан на событиях и позволяет обрабатывать несколько подключений без накладных расходов из-за переключения контекста.
- Он использует меньше памяти и ресурсов.
- NGINX делает веб-сайт быстрее и помогает им получить лучший рейтинг в Google.
- Он показывает совместимость с часто используемыми веб-приложениями, такими как ruby, python, Joomla и т.д.
- Это помогает преобразовать динамический контент в статический.
- Это помогает обрабатывать тысячи одновременных подключений одновременно.

4.2 Клиентская часть

Для создания сервера клиентской части использован Apache. Apache HTTP Server - это бесплатный веб-сервер с открытым исходным кодом, который доставляет веб-контент через Интернет.

Apache [8] - это веб-сервер, который обрабатывает запросы и обслуживает веб-ресурсы и контент по протоколу HTTP. Брандмауэры помогают защитить веб-приложение как от внешних угроз, так и от внутренних уязвимостей в зависимости от того, где настроены брандмауэры. Балансировщики нагрузки помогают распределять трафик между веб-серверами, которые обрабатывают HTTP (S) запросы (здесь на помощь приходит Apache), и серверами приложений (серверами, которые обрабатывают функциональность и рабочую нагрузку веб-приложения.) У нас также есть серверы баз данных, которые обрабатывают хранение активов и резервное копирование. В зависимости от вашей инфраструктуры ваша

база данных и приложение могут находиться на одном сервере, хотя рекомендуется хранить их отдельно.

Почему веб-серверы Apache?

Apache считается программным обеспечением с открытым исходным кодом, что означает, что исходный код находится в свободном доступе для просмотра и совместной работы. Открытый исходный код сделал Apache очень популярным среди разработчиков, которые создали и настроили свои собственные модули для применения определенных функций и улучшения его основных функций. Apache существует с 1995 года и отвечает за основную технологию, которая помогла стимулировать первоначальный рост Интернета в его зачаточном состоянии.

Одним из плюсов Apache является его способность обрабатывать большие объемы трафика с минимальной конфигурацией. Он легко масштабируется, и благодаря его модульной функциональности в основе вы можете настроить Apache так, чтобы он делал то, что вы хотите, и так, как вы этого хотите. Вы также можете удалить ненужные модули, чтобы сделать Apache более легким и эффективным.

Некоторые из наиболее популярных модулей, которые могут быть добавлены, - это SSL, Поддержка программирования на стороне сервера (PHP) и конфигурации балансировки нагрузки для обработки больших объемов трафика. Apache также может быть развернут в Linux, macOS и Windows.

Особенности веб-сервера Apache:

- Обработка статических файлов
- Загружаемые динамические модули
- Автоматическое индексирование .htaccess
- Совместимость с IPv6
- Поддерживает HTTP/2
- FTP-соединения
- Сжатие и распаковка Gzip
- Регулирование полосы пропускания
- Скрипты на Perl, PHP, Lua
- Балансировка нагрузки
- Отслеживание сеансов
- Перезапись URL-адреса
- Геолокация на основе IP-адреса

Как работает веб-сервер Apache?

Apache функционирует как способ передачи данных по сетям от клиента к серверу с использованием протокола TCP /IP. Apache может использоваться для широкого спектра протоколов, но наиболее распространенным является HTTP /S. HTTP/S или протокол передачи гипертекста (S расшифровывается как Secure) является одним из основных протоколов в Интернете, и одним из наиболее известных протоколов Apache.

HTTP/S используется для определения того, как сообщения форматируются и передаются через Интернет, с инструкциями для браузеров

и серверов о том, как отвечать на различные запросы и команды. Безопасный протокол передачи гипертекста обычно осуществляется через порт 443, а незащищенный протокол - через порт 80.

Сервер Apache настраивается с помощью конфигурационных файлов, в которых модули используются для управления его поведением. По умолчанию Apache прослушивает запрашиваемые IP-адреса, настроенные в его конфигурационных файлах. Именно здесь вступает в игру одна из многих сильных сторон Apache.

4.3 Настройка бота

Настройка бота происходит через панель администратора(Рис.4).

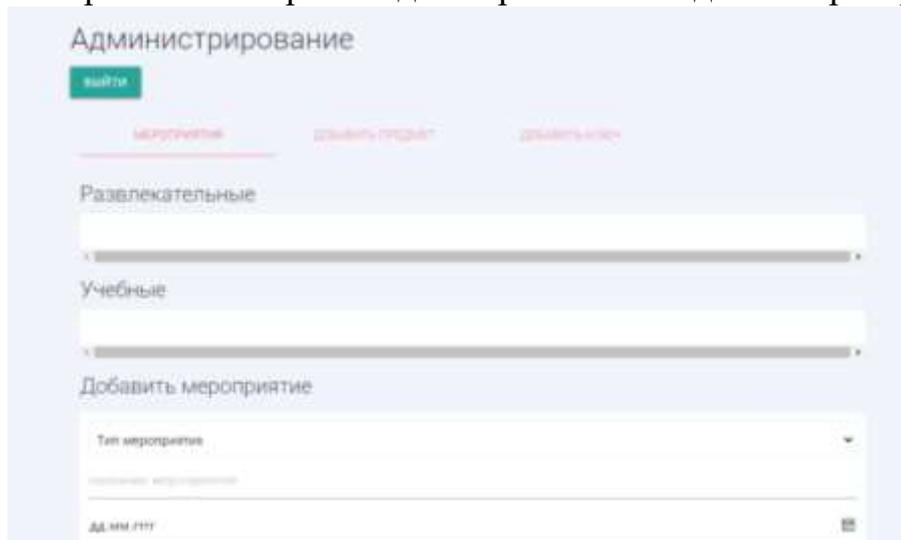


Рисунок 4 – Панель администратора

Панель администратора состоит из трех разделов: добавление мероприятия, добавление предмета, добавление ключевых слов и тегов для чат-бота.

Чтобы добавить мероприятие необходимо выбрать тип мероприятие (развлекательны, учебные), далее нужно написать информацию и дату про мероприятие, в конце нажав кнопку “Добавить”, мероприятие сразу появится в списке мероприятия и будет виден для пользователей.(Рис.5)

Рисунок 5-Добавление мероприятия

Для того, чтобы добавить учебный предмет, так же нужно заполнить информацию: название предмета, номер аудитории, данные преподавателя, оценки по предмету. После этого, нажав кнопку “Добавить”, мероприятие сразу появится в списке мероприятия и будет виден для пользователей(Рис.6).

Рисунок 6-Добавление предмета

Добавление ключей и тэгов нужно для того, чтобы бот по этим тэгам определял тип вопроса и отвечал пользователю, выдавая актуальную информацию. Для того, чтобы добавить ключ, достаточно заполнить текст информации и написать ключ для этой информации(Рис.7). При упоминании данного тэга, чат-бот будет выдавать заранее прописанный нужный текст.

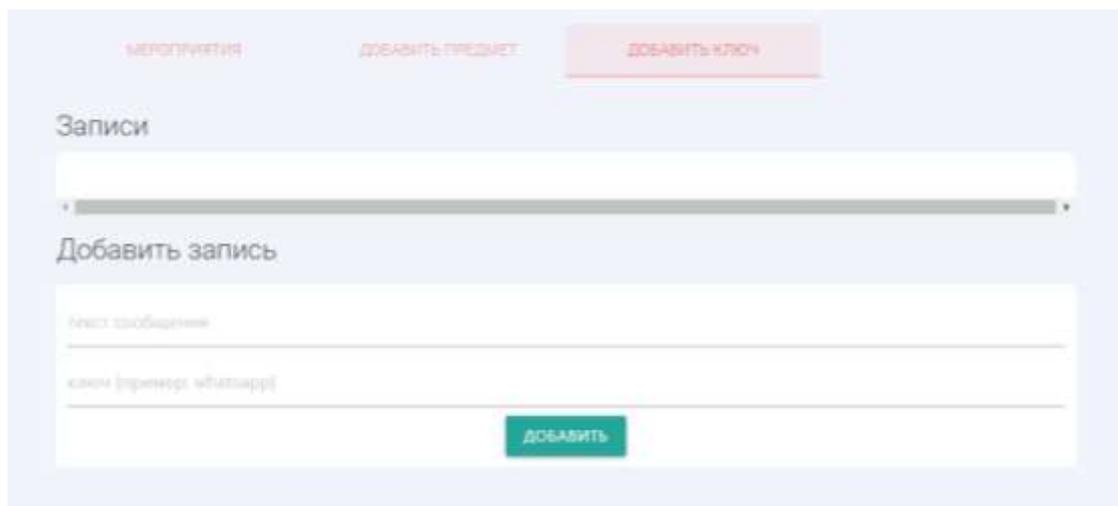


Рисунок 7-Добавление ключа

4.4 Интерфейс и тестирование чат-бота

Интерфейс чат-бота разработан с использованием React.js, интерфейсная платформа для создания одностраничных веб-приложений. Кроме того, React.js помогает в разработке адаптивных веб-страниц. Это уровень презентации, на котором пользователи (студенты / сотрудники) могут полностью взаимодействовать с чат-ботом и получать правильные и актуальные ответы. Здесь пользователь может задать боту вопросы, касающиеся текущих актуальных информации, данных преподавателей и сотрудников университета, также узнать предстоящие события и мероприятий.

Чат-боты должны быть запрограммированы так, чтобы понимать намерения клиентов. Из-за совершенно разных способов, которыми люди сообщают об индивидуальных намерениях, для представления входных значений, которые чат-боты должны уметь обрабатывать, требуются значительные данные.

Эти обучающие данные представляют собой набор сценариев, выражающих различные виды намерений. Затем он должен быть протестирован, чтобы убедиться, что его реакция соответствует ожиданиям в каждом сценарии. Наш чат-бот успешно прошел несколько этапов тестирования и готов к эксплуатации

5. РЕАЛИЗАЦИЯ ЧАТ-БОТА В ИНФОРМАЦИОННОЙ ОБРАЗОВАТЕЛЬНОЙ СИСТЕМЕ

Пользователи могут взаимодействовать с чат – ботом несколькими образами: используя кнопки с параметрами или встроенную клавиатуру.Так же добавлен меню [10] для удобства пользователей с вкладками(см. Рис. 8):

- Данные о предмете
- Добавить/Убрать предмет
- Мероприятия
- Написать

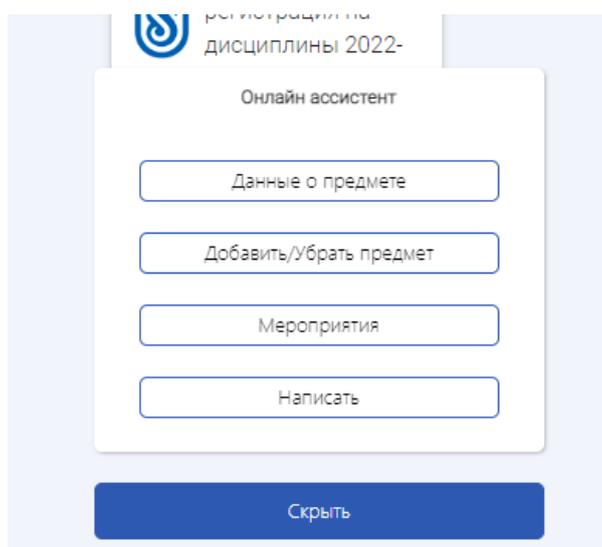


Рисунок 8 – Меню чат-бота

Для начала диалога с чат-ботом нужно нажать кнопку «Написать», далее задать интересующий вопрос(см. Рис.9,10,11).

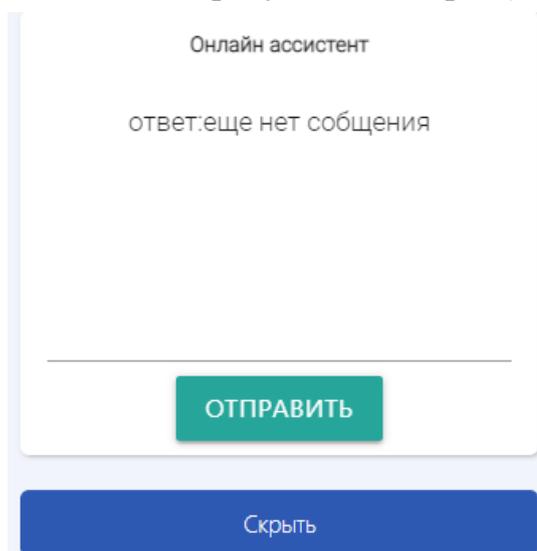


Рисунок 9 – Начало диалога с онлайн-ассистентом

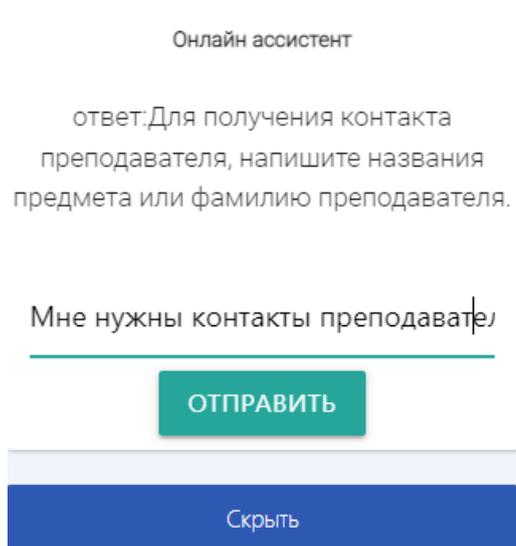


Рисунок 10 – Вопрос Чат-Боту

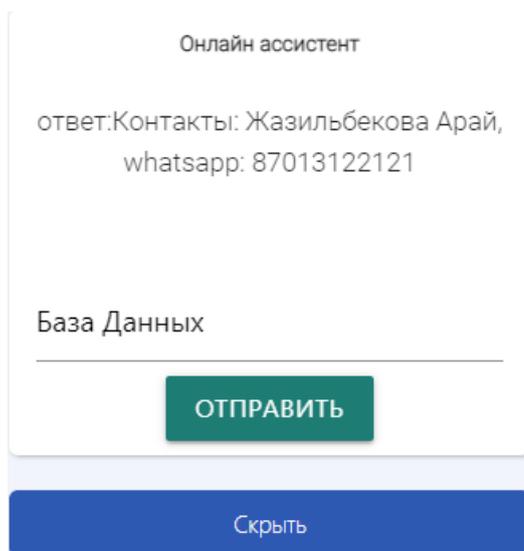


Рисунок 11 – Получение ответа от Чат-Бота

ЗАКЛЮЧЕНИЕ

В данной дипломной работе рассматриваются вопросы исследования и цель, а также обобщаются найденные ответы и ключевые моменты обсуждения. Также представлены и обсуждены будущие исследования.

Целью дипломной работы было дальнейшее развитие образовательных чат-ботов путем анализа того, что было сделано, и обобщения этих знаний. Это можно было бы сделать, выяснив, какими возможностями может обладать чат-бот в образовательном контексте и может ли он функционировать самостоятельно или ему требуются дополнительные технологии для повышения педагогической ценности образования. Эта цель сводилась к следующим открытым вопросам: Какие роли могут играть чат-боты образовательном контексте? И какими функциями могут обладать чат-боты?.

На этот вопрос был дан исчерпывающий ответ в этой работе, но ответ можно резюмировать следующим образом: функций и применений чат-ботов много, и они, похоже, ограничены только тем, что мы, люди, можем себе представить, что они делают. Это гибкий и разнообразный инструмент, одновременно простой и сложный, ограниченный только изобретательностью его создателя. Он может хорошо работать сам по себе и обеспечивают образовательную ценность, но могут достигать еще лучших результатов в сочетании с другими технологиями.

Более конкретный ответ может заключаться в том, что чат-бот может использоваться в качестве репетитора, оценщика учащихся, для вопросов и ответов, для общения с учителем или просто для естественной беседы. Возможности применения этих ролей кажутся бесконечными, полезные функций указанные в этой работе, скорее всего, являются лишь подмножеством того, что чат-боты действительно могут делать. Возможности чат-ботов можно расширить, включив их в другие системы, такие как системы электронного обучения, виртуальная сред или библиотечной системы (или других систем с большим объемом баз данных) или путем добавления дополнительных технологий, таких как технология преобразования текста в речь, лингвистические инструменты или анимации.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. What is REST [Электронный ресурс] –
[\[https://restfulapi.net/\]](https://restfulapi.net/)
2. Top Features and Benefits of Using React JS for Web Development [Электронный ресурс] –
[\[https://webandcrafts.com/blog/react-js-features/\]](https://webandcrafts.com/blog/react-js-features/)
3. Advantages and Disadvantages of MongoDB NoSQL Database [Электронный ресурс] –
[\[https://webandcrafts.com/blog/mongodb-advantages-and-disadvantages/\]](https://webandcrafts.com/blog/mongodb-advantages-and-disadvantages/)
4. The Good and the Bad of Node.js Web App Development [Электронный ресурс]–
[\[https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-node-js-web-app-development/\]](https://www.altexsoft.com/blog/engineering/the-good-and-the-bad-of-node-js-web-app-development/)
5. StackOverFlow [Электронный ресурс] –
[\[https://ru.stackoverflow.com/questions/tagged/node.js\]](https://ru.stackoverflow.com/questions/tagged/node.js)
6. WebStorm The smartest JavaScript IDE [Электронный ресурс] –
[\[https://www.jetbrains.com/webstorm/\]](https://www.jetbrains.com/webstorm/)
7. Introduction to JSON Web Tokens [Электронный ресурс] –
[\[https://jwt.io/introduction\]](https://jwt.io/introduction)
8. What is apache [Электронный ресурс] –
[\[https://www.hostinger.com/tutorials/what-is-apache\]](https://www.hostinger.com/tutorials/what-is-apache)
9. How to Build AI Chatbots for Your Clients [Электронный ресурс] –
[\[https://www.shopify.com/partners/blog/ai-chatbots\]](https://www.shopify.com/partners/blog/ai-chatbots)
10. How To Build a Menu Based Chatbot To Increase Site Conversions by 107% [Электронный ресурс] –
[\[https://mobilemonkey.com/blog/menu-based-chatbot\]](https://mobilemonkey.com/blog/menu-based-chatbot)
11. What is Apache? In-Depth Overview of Apache Web Server [Электронный ресурс] –
[\[https://www.sumologic.com/blog/apache-web-server-introduction/\]](https://www.sumologic.com/blog/apache-web-server-introduction/)
12. A Simple Chatbot Solution - Why You Should Use Button Bots [Электронный ресурс] –
[\[https://www.giosg.com/blog/why-you-should-use-button-bots\]](https://www.giosg.com/blog/why-you-should-use-button-bots)
13. Express [Электронный ресурс] –
[\[https://www.npmjs.com/package/express\]](https://www.npmjs.com/package/express)
14. Mongoose [Электронный ресурс] –
[\[https://mongoosejs.com/\]](https://mongoosejs.com/)
15. What is Axios.js and why should I care? [Электронный ресурс] –
[\[https://medium.com/@MinimalGhost/what-is-axios-js-and-why-should-i-care-7eb72b111dc0\]](https://medium.com/@MinimalGhost/what-is-axios-js-and-why-should-i-care-7eb72b111dc0)
16. What Is Nginx? A Basic Look at What It Is and How It Works [Электронный ресурс] –
[\[https://kinsta.com/knowledgebase/what-is-nginx/\]](https://kinsta.com/knowledgebase/what-is-nginx/)

Приложение А

(обязательное)

Техническое задание

А1.1 Техническое задание на разработку информационной системы чат-бота.

Данное техническое задание распространяется на проектирование и разработку информационной системы для чат-бота. Автоматизирование информационной системы чат-бота позволит облегчить взаимодействие между преподавателем и студентом.

А1.1.1 Основания для разработки

Система разрабатывается на основании моего личной инициативы как студента и пользователя образовательной системы, для улучшения взаимодействия отношений преподавателя и студента.

А.1.1.2 Назначение

Разрабатываемый проект предназначен для улучшения качества образования в университете.

Продолжение приложения А

А.1.1.3 Требования к функциональным характеристикам

Система должна обеспечить возможность выполнения следующих функций:

- Add/Drop предмета
- Получение обратной связи от студентов о предмете
- Информация об успеваемости
- предоставление информации о мероприятиях
- Онлайн-Ассистент с возможностью ответа по ключевому слову
- Корректировка данных о Мероприятиях через БД
- Корректировка данных о предмете через БД
- авторизация сотрудников и студентов
- предоставление доступа сотрудникам к определённым данным

А.1.1.4 Требования к надежности

Предоставление конфиденциальности университетских данных для преподавательского состава. Ограничение возможностей действий для студентов. Индивидуальная авторизация для каждого человека.

А.1.1.5 Требования к составу и параметрам технических средств

Система должна работать на всех персональных компьютерах, смартфонах и различных браузерах. Минимальные требования: тип процессора – Windows XP и выше, оперативная память – 64 Мб RAM и выше.

А.1.1.6 Требования к информационной и программной совместимости

Система должна работать под управлением любой операционной системы, так как она является кроссплатформенной.

Приложение Б

(обязательное)

Текст программы

```
// код hooks.js:
import {useState, useCallback} from 'react'

export const useHttp = () => {
  const [loading, setLoading] = useState(false)
  const [error, setError] = useState(null)

  const request = useCallback(async (url, method = 'GET', body = null,
headers = {}) => {
    setLoading(true)
    try {
      if (body) {
        body = JSON.stringify(body)
        headers['Content-Type'] = 'application/json'
      }

      const response = await fetch(url, {method, body, headers})
      const data = await response.json()

      if (!response.ok) {
        throw new Error(data.message || 'Что-то пошло не так')
      }

      setLoading(false)
      return data
    } catch (e) {
      setError(e)
      setLoading(false)
    }
  })
}
```

Продолжение приложения Б

```
setError(e.message)
  throw e
}
}, [])

const clearError = useCallback(() => setError(null), [])

return { loading, request, error, clearError }
}

// код модели для мероприятий

const {Schema, model.Types} = require('mongoose')

const schema = new Schema({
  name: {type:String},
  date: {type:String},
  place: {type:String},
  type: {type:String},
  state: {type:String}
})

module.exports = model('Event', schema)

// код модели для предмета

const {Schema, model.Types} = require('mongoose')

const schema = new Schema({
  state:{type:String},
  name: {type:String},
  mark: {type:String},
  contact: {type:String},
  officehours:{type:String}
})
```

Продолжение приложения Б

```
setError(e.message)
  throw e
}
}, [])

const clearError = useCallback(() => setError(null), [])

return { loading, request, error, clearError }
}

// код модели для мероприятий

const {Schema, model.Types} = require('mongoose')

const schema = new Schema({
  name: {type:String},
  date: {type:String},
  place: {type:String},
  type: {type:String},
  state: {type:String}
})

module.exports = model('Event', schema)

// код модели для предмета

const {Schema, model.Types} = require('mongoose')

const schema = new Schema({
  state:{type:String},
  name: {type:String},
  mark: {type:String},
  contact: {type:String},
  officehours:{type:String}
})
```

Протокол анализа Отчета подобия заведующего кафедрой

Заведующий кафедрой заявляет, что ознакомился (-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Абдуханов Олжас

Название: Диплом_Абдуханов_Олжас.docx

Координатор: Косников Вячеслав Алексеевич

Коэффициент подобия 1: 4,59%

Коэффициент подобия 2 : 2,71

После анализа Отчета подобия заведующий кафедрой констатирует следующее:

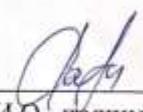
обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, работа признается самостоятельной и допускается к защите;

обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы, по существу, и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;

обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, работа не допускается к защите.

20 мая 2022 г.

Дата/м./г.



Ф.И.О., подпись зав.кафедрой

Отзыв научного руководителя

На дипломный проект студента специальности 5В07030
«Информационные системы» Абдуханова Олжаса Аскарулы
«Разработка и реализация системы «Chatbot» для автоматизации
взаимодействия между преподавателем и студентом в университете»

Целью данного дипломного проекта является разработка чат-бот системы для автоматизации отношений между преподавателем и студентом, которая решает проблему простоты интеграции в различные цифровые платформы без необходимости загрузки приложения, сокращает время затрачиваемое на организацию и выполнение задач, поскольку чат-боты дают немедленные ответы, заранее спроектированные, на частые вопросы студентов. Можно отметить, что актуальность поставленной цели не вызывает сомнений.

В первой главе дипломного проекта произведен анализ предметной области и поставлены задачи проектирования и разработки чат-бота.

Вторая глава посвящена выбору инструментальных средств и технологий проектирования системы.

В третьей главе произведен сбор и анализ необходимых данных.

Четвертая глава посвящена Процессу реализации системы.

При выполнении проекта Абдуханов Олжас Аскарулы показал высокий уровень теоретической и практической подготовки. Все поставленные задачи полностью решены. Считаю, что проект Абдуханова Олжаса Аскарулы может быть допущен к защите и заслуживает высокой оценки.

Научный руководитель
канд.техн.наук, доцент



Косников В.А.

РЕЦЕНЗИЯ

на дипломный проект студента 4 курса Казахского национального
исследовательского технического университета им.К.И.Сатпаева
специальности 5В070300 «Информационные системы»

Абдуханов Олжас Аскаруды

на тему: «Разработка и реализация системы «Chatbot» для
автоматизации взаимодействия между преподавателем и
студентом в университете»

Структура дипломной работы включает в себя: введение, пять разделов, заключение, список используемых источников литературы и приложений, а также 11 рисунков.

Во введении описываются применение и функции информационной системы чат-бота.

В первой главе дипломной работы анализируется предметная область, ставятся задачи и описывается применение чат-бота в образовательной системе.

Во второй главе рассматриваются используемые инструменты и технологии, определяются преимущества и недостатки программной платформы и проводится сравнительный анализ с другими средами разработки.

В третьей главе собирается информация необходимая для создания и полного функционирования чат-бота. Загружается и строится структура запросов. Анализ полученной информации и выбор базы данных.

В четвертой главе реализуется чат-бот с серверной и клиентской стороны. Настройка внутренних функции онлайн-ассистента.

В пятой главе проводится реализация чат-бота в информационно-образовательной системе.

В заключении приведены выводы о проделанной работе.

Замечание по проекту:

В целом работа представлена завершенной и может быть оценена на «отлично», а при успешной защите Абдуханов О. достоин присвоения академической степени бакалавра технических наук.

Рецензент:

доктор PhD, доцент

Казахстанско-немецкого университета



O.V.

Киселева О.В.