МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение
высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Центр проектной деятельности

Промежуточный отчёт №1

по дисциплине «Проектная деятельность»

Тема проекта:

«EasyAccess. Браузерное расширение для повышения доступности веб-сайтов»

Проект выполняют:

1. Остапенко Владислав Русланович	8. Пахалюк Илья Николаевич
2. Деев Егор Викторович	9. Петрачков Владимир Владимирович
3. Момина Антонина Алексеевна	10. Поправкин Александр Валентинович
4. Мясников Дмитрий Сергеевич	11. Сапрыкин Петр Иванович
5. Николенко Дарья Романовна	12. Старков Руслан Владимирович
6. Ночной Максим Сергеевич	13. Трошкин Дмитрий Александрович
7. Остапенко Святослав Русланович	14. Шмыговский Никита Сергеевич

Руководители проекта:

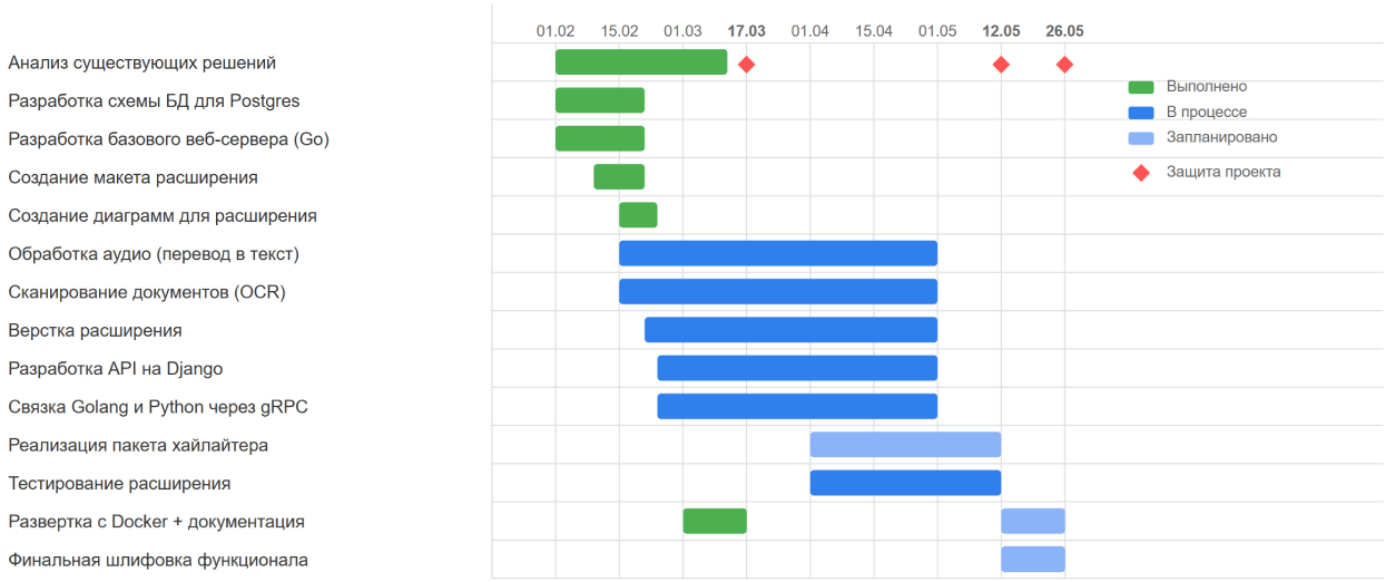
(подпись) / Будылина Е. А. к.т.н., доцент

(подпись) / Киреева Г. И. к.т.н., доцент

2025

СОДЕРЖАНИЕ

ГРАФИК РАБОТ



ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

API (Application Programming Interface) — набор правил и протоколов, который позволяет различным компонентам приложения взаимодействовать друг с другом.

CSS (Cascading Style Sheets) — язык стилей, используемый для описания внешнего вида и форматирования HTML-документов.

DOM (Document Object Model) — программный интерфейс для HTML и XML документов, представляющий их в виде древовидной структуры.

gRPC (Google Remote Procedure Call) — высокопроизводительный фреймворк для удаленного вызова процедур, разработанный Google.

HTML (HyperText Markup Language) — язык разметки, используемый для создания и структурирования веб-страниц.

IDE (Integrated Development Environment) — интегрированная среда разработки для создания программного обеспечения.

JSON (JavaScript Object Notation) — текстовый формат обмена данными, основанный на JavaScript.

OCR (Optical Character Recognition) — технология, позволяющая преобразовывать различные типы документов в редактируемые и поддающиеся поиску данные.

REST (Representational State Transfer) — архитектурный стиль взаимодействия компонентов распределённого приложения в сети.

SQL (Structured Query Language) — язык программирования для работы с реляционными базами данных.

СУБД (Система управления базами данных) — программное обеспечение для создания, хранения и управления базами данных.

UML (Unified Modeling Language) — язык графического описания для объектного моделирования.

UUID (Universally Unique Identifier) — стандарт идентификации, используемый в создании программного обеспечения.

ВВЕДЕНИЕ

В современном мире интернет стал неотъемлемой частью повседневной жизни, а веб-приложения и сайты – основным инструментом для работы, развлечений и коммуникации. Однако функциональность и интерфейс многих веб-ресурсов не всегда соответствуют потребностям конкретных пользователей, особенно людей с ограниченными возможностями здоровья. Возникает необходимость в инструментах, позволяющих модифицировать и расширять возможности существующих веб-сайтов для обеспечения их доступности и адаптации под индивидуальные потребности пользователей.

Проект EasyAccess представляет собой комплексное браузерное расширение, разработанное для универсальной модификации веб-страниц с целью повышения доступности контента, отраженного на сайте. Приложения предназначено для лиц с ограниченными возможностями здоровья.

EasyAccess объединяет в себе возможности нескольких специализированных расширений, таких как Tampermonkey (для выполнения пользовательских скриптов) и Stylus (для применения пользовательских стилей), и предоставляет дополнительные функции, включая работу с документами и распознавание голоса.

Целью проекта является разработка универсального браузерного расширения EasyAccess, способного повышать доступность веб-сайтов и добавлять к ним новую функциональность через систему пакетов модификаций. Расширение должно предоставлять возможности для настройки визуального отображения, голосового управления, обработки документов и других функций, важных для пользователей с различными ограничениями.

Актуальность проекта обусловлена растущей потребностью в инструментах, повышающих доступность веб-ресурсов для лиц с ограниченными возможностями здоровья. Особую значимость такие инструменты приобретают в контексте государственных и социально значимых веб-ресурсов, которые должны быть доступны для всех категорий пользователей.

Для достижения поставленной цели были определены следующие задачи:

1. Провести анализ существующих решений по обеспечению доступности веб-ресурсов.
2. Разработать архитектуру расширения, позволяющую гибко настраивать и расширять его функциональность.
3. Спроектировать и реализовать базу данных для хранения пакетов модификаций и пользовательских настроек.
4. Разработать серверную инфраструктуру для обеспечения работы расширения.
5. Создать систему пакетов модификаций для повышения доступности, позволяющую управлять стилями и скриптами для различных веб-сайтов.
6. Реализовать модули для сканирования документов и обработки аудио файлов.
7. Обеспечить удобный пользовательский интерфейс, соответствующий принципам доступности.
8. Разработать веб-редактор для создания пакетов модификаций.
9. Подготовить инфраструктуру для развертывания проекта с использованием Docker.
10. Сформировать документацию проекта и поддерживать её актуальность.

Объектом исследования являются методы и технологии повышения доступности веб-ресурсов и расширения их функциональности для различных категорий пользователей, включая людей с ограниченными возможностями здоровья.

Предметом исследования выступают конкретные технические решения для реализации универсального браузерного расширения, способного применять различные пакеты модификаций к веб-страницам для повышения их доступности.

В процессе выполнения проекта были использованы различные методы и подходы, включая объектно-ориентированное программирование, реляционные базы данных, а также современные веб-технологии для создания доступных интерфейсов.

Практическая значимость проекта заключается в создании универсального инструмента, который может быть использован для решения широкого спектра задач по повышению доступности веб-ресурсов:

- изменение внешнего вида веб-сайтов для пользователей с нарушениями зрения (настройка контрастности, размера текста, цветовых схем);
- улучшение навигации для пользователей с двигательными нарушениями;
- добавление альтернативного текста к изображениям для пользователей с нарушением зрения;
- распознавание и озвучивание текста для незрячих пользователей;
- голосовое управление для пользователей с ограниченными возможностями самостоятельного ввода;
- работа с документами и извлечение из них информации;
- персонализация интерфейсов для пользователей с когнитивными нарушениями.

EasyAccess может найти применение как среди индивидуальных пользователей с особыми потребностями, так и в организациях, стремящихся обеспечить доступность своих веб-ресурсов для всех категорий пользователей.

1 ПРЕДПРОЕКТНОЕ ИССЛЕДОВАНИЕ

1.1 Описание предметной области

Веб-доступность (Web Accessibility) и модификация веб-страниц представляют собой две взаимосвязанные области, на которых фокусируется проект EasyAccess.

Веб-доступность

Веб-доступность подразумевает создание веб-сайтов и приложений, которые могут использоваться всеми людьми, независимо от их физических возможностей, сенсорных способностей или технической оснащенности. Это особенно важно для людей с различными видами ограничений.

Например, нарушения зрения – от полной слепоты до различных форм дальтонизма и частичной потери зрения. Пользователи с такими ограничениями могут использовать программы экранного доступа (скринридеры), увеличение экрана или настройку контрастности.

Нарушения слуха – пользователи с нарушениями слуха нуждаются в текстовых альтернативах для аудиоконтента, таких как субтитры или расшифровки.

Двигательные нарушения – люди с ограниченной подвижностью могут испытывать трудности при использовании мыши и нуждаются в полной доступности сайта с клавиатуры или альтернативных устройств ввода.

Когнитивные нарушения – включают проблемы с обучением, концентрацией внимания или пониманием сложной информации. Для таких пользователей важно иметь четкую и простую структуру контента.

По данным Всемирной организации здравоохранения, более 1 миллиарда людей в мире живут с той или иной формой инвалидности. В России, согласно официальной статистике, насчитывается около 11,9 миллионов людей с ограниченными возможностями. Все эти люди могут испытывать трудности при использовании веб-ресурсов, если они не адаптированы должным образом.

Существуют международные стандарты и рекомендации по обеспечению веб-доступности, такие как WCAG (Web Content Accessibility Guidelines), разработанные W3C. В Российской Федерации вопросы доступности веб-ресурсов регулируются ГОСТ Р 52872-2019 «Интернет-ресурсы и другая информация, представленная в электронно-цифровой форме. Приложения для стационарных и мобильных устройств, иные пользовательские интерфейсы. Требования доступности для людей с инвалидностью и других лиц с ограничениями жизнедеятельности».

Модификация веб-страниц

Модификация веб-страниц через браузерные расширения представляет собой технологию, позволяющую изменять содержимое, внешний вид и функциональность веб-страниц без изменения их исходного кода на стороне сервера. Это достигается путем внедрения пользовательских стилей CSS и скриптов JavaScript, которые переопределяют или дополняют оригинальный код страницы.

Браузерные расширения для модификации веб-страниц предлагают различные функции:

- изменение визуального оформления (стили CSS);
- добавление новой функциональности (скрипты JavaScript);
- изменение содержимого страницы (манипуляция DOM);
- автоматизация действий на сайте;
- интеграция с внешними сервисами.

Объединение подходов в EasyAccess

Проект EasyAccess объединяет концепции веб-доступности и модификации веб-страниц через систему пакетов модификаций – комбинаций стилей CSS, скриптов JavaScript и настроек, которые могут быть применены к определенным веб-сайтам для повышения их доступности.

Пакет модификаций в EasyAccess включает:

- CSS-компонент – стили для изменения внешнего вида сайта, повышения контрастности, изменения размеров текста и других визуальных аспектов.
- JavaScript-компонент – скрипты для изменения функциональности, добавления поддержки клавиатурной навигации, интеграции с программами экранного доступа.
- JSON-компонент – стили, сгенерированные во внутреннем веб-редакторе расширения.
- Метаданные – информация о пакете, включая название, описание, автора и версию.

Базовая функциональность EasyAccess в контексте доступности включает:

1. Настройка визуального отображения – изменение размера текста, контрастности, цветовых схем.
2. Голосовое управление – распознавание голосовых команд, чтение контента вслух.
3. Работа с документами – распознавание текста из изображений и документов.
4. Клавиатурная навигация – улучшение доступности управления с клавиатуры.
5. Упрощение интерфейса – удаление отвлекающих элементов для улучшения восприятия.

Особенностью EasyAccess является возможность создания специализированных пакетов модификаций для конкретных веб-сайтов и сервисов, что позволяет более точно адаптировать их под потребности пользователей с ограниченными возможностями. Например, могут быть созданы специальные пакеты для государственных порталов, образовательных ресурсов или популярных социальных сетей.

Универсальность платформы EasyAccess позволяет расширять её функциональность через создание новых пакетов модификаций без изменения основного кода, что делает её гибким и масштабируемым решением для повышения доступности веб-ресурсов.

1.2 Сравнительный анализ программ-аналогов

Для оценки конкурентоспособности и определения уникальных характеристик разрабатываемого расширения EasyAccess был проведен сравнительный анализ существующих аналогов. В качестве объектов для анализа были выбраны как специализированные решения в области веб-доступности, так и универсальные расширения для модификации веб-страниц.

Объекты анализа:

1) Tampermonkey – популярное расширение для управления пользовательскими скриптами JavaScript, позволяющее изменять функциональность веб-страниц.
2) Stylus – расширение для создания и применения пользовательских стилей CSS, дающее возможность изменять внешний вид сайтов.
3) Accessibly – специализированное расширение для повышения доступности веб-страниц, включающее базовые настройки отображения.
4) Screen Reader – расширение, интегрирующееся с технологиями экранного доступа для улучшения восприятия веб-контента пользователями с нарушениями зрения.
5) Версии для слабовидящих – встроенные функции на государственных и социально значимых сайтах РФ, обеспечивающие альтернативное отображение контента.

Критерии сравнения:

- поддержка пользовательских JavaScript-скриптов;
- поддержка пользовательских CSS-стилей;
- настройка размера текста и контрастности;

- голосовое управление и поддержка голосового ввода;
- работа с документами и распознавание текста;
- система организации и управления модификациями;
- маркетплейс/репозиторий для обмена модификациями;
- интеграция с технологиями экранного доступа;
- комплексные пакеты модификаций (CSS+JS+JSON);
- открытость исходного кода;
- поддержка русского языка.

Результаты анализа представлены в таблице 1.

Таблица 1. Сравнительный анализ аналогов

Функциональные возможности	**Tampermonkey**	**Stylus**	**Accessibility**	**Screen Reader**	**Версии для слабовидящих**	**EasyAccess**
Пользовательские JavaScript-скрипты	+	-	-	+/-	-	+
Пользовательские CSS-стили	-	+	+/-	-	+	+
Настройка размера текста и контрастности	-	-	+	-	+	+
Голосовое управление	-	-	-	+/-	-	+
Работа с документами и распознавание текста	-	-	-	-	-	+
Система организации модификаций	+	+	-	-	-	+

Продолжение таблицы 1

Функциональные возможности	**Tampermonkey**	**Stylus**	**Accessibility**	**Screen Reader**	**Версии для слабовидящих**	**EasyAccess**
Маркетплейс/репозиторий	+	+	-	-	-	+
Интеграция с технологиями экранного доступа	-	-	+	+	+/-	+
Комплексные пакеты модификаций	-	-	-	-	-	+
Поддержка русского языка	+	+	+/-	+/-	+	+
Открытый исходный код	-	+	+/-	-	-	+

Осуществлённый анализ позволяет сделать следующие выводы:

1. Специализация vs. универсальность: Существующие решения либо специализируются на отдельных аспектах модификации веб-страниц (Tampermonkey, Stylus), либо на ограниченном наборе функций веб-доступности (Accessibility, Screen Reader). Ни одно из них не предоставляет комплексного решения, объединяющего оба направления.
2. Ограниченная функциональность: Решения в области веб-доступности часто предлагают лишь базовые настройки (размер текста, контрастность), не обеспечивая глубокой модификации страниц или интеграции с внешними сервисами.

3. Отсутствие комплексных пакетов: Ни один из аналогов не предлагает системы пакетов, объединяющих CSS, JavaScript и конфигурацию в единый набор для комплексной модификации сайтов.
4. Недостаточное внимание к русскоязычному сегменту: Многие специализированные решения для доступности имеют ограниченную поддержку русского языка и плохо адаптированы к особенностям российских веб-ресурсов.
5. Версии для слабовидящих: Реализованные на российских государственных сайтах версии для слабовидящих часто имеют ограниченную функциональность и работают только в рамках конкретного сайта, не предоставляя универсального решения.

Уникальные преимущества, которые будут реализованы в проекте EasyAccess:

1. Универсальный подход: Объединение функциональности для модификации веб-страниц и повышения доступности в едином решении.
2. Система пакетов модификаций: Возможность создания, управления и обмена комплексными пакетами, включающими стили, скрипты и настройки.
3. Расширенная функциональность: Интеграция возможностей для работы с документами, распознавания текста и голосового управления.
4. Фокус на российский рынок: Полная поддержка русского языка и особое внимание к потребностям пользователей российских веб-ресурсов.
5. Открытая платформа: Открытый исходный код и архитектура, позволяющая сообществу разработчиков расширять функциональность и создавать новые пакеты модификаций.

В результате анализа подтверждена актуальность разработки EasyAccess как комплексного решения, объединяющего возможности существующих

специализированных расширений и предлагающего дополнительную функциональность для повышения доступности веб-ресурсов.

1.3 Постановка задачи

На основе проведенного анализа предметной области и существующих аналогов была выполнена постановка задачи для разработки браузерного расширения EasyAccess.

Цель проекта – разработать универсальное браузерное расширение, объединяющее возможности модификации веб-страниц и повышения их доступности через систему пакетов модификаций, с особым вниманием к потребностям пользователей с ограниченными возможностями и спецификам российских веб-ресурсов.

Базовые функциональные требования:

1. Система пакетов модификаций:
    - создание и управление пакетами, включающими CSS, JavaScript и JSON-конфигурацию,
    - селективное применение пакетов на основе URL-фильтров,
    - версионирование пакетов и обновления,
    - маркетплейс для обмена пакетами;
2. Повышение доступности:
    - настройка размера текста, контрастности и цветовых схем,
    - поддержка навигации с клавиатуры,
    - интеграция с технологиями экранного доступа,
    - упрощение интерфейса для улучшения восприятия,
    - специализированные пакеты для популярных российских веб-ресурсов;
3. Модификация веб-страниц:
    - инъекция пользовательских CSS-стилей,
    - выполнение пользовательских JavaScript-скриптов,
    - изменение структуры страницы (DOM),

- автоматизация действий на сайтах;

4. Дополнительные функции, реализуемые в виде предустановленных пакетов:
   - сканирование и распознавание текста из изображений и документов
   - голосовое управление и ввод текста
   - выделение и сохранение текста с защищенных от копирования страниц
   - работа с иерархическими структурами данных.

Технические требования:

1. Архитектура:
   - Frontend: TypeScript, React для разработки пользовательского интерфейса маркетплейса пакетов;
   - Backend: Go для серверной части и работы с базой данных, Python для скриптов обработки данных;
   - База данных: PostgreSQL (TimescaleDB) для хранения пакетов модификаций и пользовательских настроек.
2. Расширение браузера:
   - Typescript, Solid.js для разработки пользовательского интерфейса расширения;
   - Совместимость с Chromium-based браузерами (Chrome, Edge, Yandex);
   - Модульная структура с возможностью включения/отключения компонентов;
   - Минимальное потребление ресурсов браузера;
   - Возможность работы в автономном режиме.
3. Интеграция:

- Использование gRPC для связи между компонентами на разных языках программирования;
- Интеграция с API для распознавания голоса и текста;
- Возможность взаимодействия с внешними сервисами через REST API.

4. Развертывание:
   - Использование Docker для контейнеризации и упрощения развертывания серверной части;
   - Обеспечение возможности локальной установки для корпоративных клиентов;
   - Система автоматического обновления расширения и пакетов модификаций.
5. Безопасность:
   - Надежное хранение пользовательских данных;
   - Проверка скриптов на потенциально вредоносный код;
   - Разграничение доступа к функциям расширения;
   - Защита каналов передачи данных.
6. Веб-редактор:
   - Интерфейс для создания и редактирования пакетов модификаций;
   - Визуальный редактор CSS с предпросмотром изменений;
   - Редактор JavaScript с подсветкой синтаксиса и автодополнением;
   - Система управления версиями пакетов.

1.4 Характеристика инструментальных средств

Для реализации проекта EasyAccess был выбран комплекс современных инструментальных средств, обеспечивающих эффективную разработку, тестирование и развертывание всех компонентов системы.

Языки программирования

1. Typescript – строго типизированная надстройка над JavaScript, используется для разработки фронтенд-части расширения и веб-редактора. Основные преимущества:
   - статическая типизация, снижающая количество ошибок при разработке,
   - улучшенный инструментарий и поддержка IDE,
   - обратная совместимость с JavaScript,
   - поддержка современных конструкций ECMAScript.
2. Go (Golang) – современный компилируемый язык, применяется для разработки серверной части и взаимодействия с базой данных. Ключевые особенности:
   - высокая производительность и эффективность,
   - встроенная поддержка конкурентного программирования,
   - строгая типизация и простота синтаксиса,
   - быстрая компиляция и отличный инструментарий.
3. Python – применяется для создания скриптов обработки данных, распознавания голоса и сканирования документов:
   - богатая экосистема библиотек для машинного обучения и обработки данных,
   - простота интеграции с внешними API,
   - высокая скорость разработки и читаемость кода,
   - широкое сообщество разработчиков.
4. HTML/CSS – стандартные языки разметки и стилей для веб-разработки:
   - универсальность и совместимость с различными платформами,
   - возможность создания отзывчивых и адаптивных интерфейсов,
   - доступность и соответствие стандартам W3C,
   - обширная документация и поддержка сообщества.

Фреймворки и библиотеки

1. React – библиотека JavaScript для построения пользовательских интерфейсов:
   - компонентный подход к разработке,
   - виртуальный DOM для оптимизации рендеринга,
   - однонаправленный поток данных,
   - богатая экосистема дополнительных библиотек.
2. Solid.js – более легковесная версия React с меньшим временем первой отрисовки.
3. Chrome Extension API – набор API для создания расширений браузера:
   - доступ к функциям браузера и вкладкам,
   - возможность модификации контента веб-страниц,
   - управление хранилищем данных расширения,
   - коммуникация между различными частями расширения.
4. Django – высокоуровневый Python-фреймворк для разработки веб-приложений:
   - встроенный ORM для работы с базами данных,
   - система шаблонов для генерации HTML,
   - административный интерфейс,
   - безопасность и масштабируемость.

Системы управления базами данных

1. PostgreSQL — свободная объектно-реляционная СУБД:
   - надежность и соответствие стандартам SQL,
   - расширяемость через системы плагинов,
   - поддержка JSON/JSONB для хранения структурированных данных,
   - мощные возможности индексирования и оптимизации запросов.

2. TimescaleDB – расширение для PostgreSQL, оптимизированное для временных рядов:
    - эффективное хранение и запрос временных данных,
    - автоматическое партиционирование по времени,
    - расширенные функции агрегации для временных рядов,
    - полная совместимость с PostgreSQL.

Выбор данных инструментальных средств обусловлен их соответствием требованиям проекта, современностью, надежностью и наличием активного сообщества разработчиков. Комбинация этих технологий позволяет создать эффективное, масштабируемое и удобное в поддержке браузерное расширение, способное решать широкий спектр задач по модификации веб-страниц и повышению их доступности.

2 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ ВЕБ-СЕРВИСА

2.1 Проектирование программного приложения (Трошкин Д., Деев Е., Шмыговский Н.)

Разрабатываемое в рамках проекта браузерное расширение EasyAccess относится к категории инструментов повышения доступности веб-сайтов, предназначенных для обеспечения комфортной работы с интернет-ресурсами людям с ограниченными возможностями здоровья. Архитектура приложения предусматривает использование нескольких компонентов для обеспечения полной функциональности.

Наш проект состоит из нескольких ключевых технических блоков. На стороне пользователя работает расширение для браузера, написанное на TypeScript с использованием библиотек React для интерфейса. Серверная часть разделена на два основных компонента: быстрый сервис на языке Go, который взаимодействует с базой данных, и набор Python-модулей, отвечающих за анализ и обработку данных. Дополнительно реализованы специализированные компоненты для работы с документами различных форматов и распознавания речи.

Эта многокомпонентная архитектура позволяет эффективно разделить функциональность, обеспечить безопасность и повысить производительность приложения.

Передача информации между клиентом и сервером осуществляется посредством HTTP-протоколов и gRPC для обмена структурированными данными. Клиентское расширение отправляет запросы на API-сервер для получения пакетов модификаций, обработки настроек пользователя и других операций.

База данных проекта построена на PostgreSQL - популярной реляционной СУБД, дополненной расширением TimescaleDB. Это расширение специально оптимизировано для эффективного хранения и обработки данных с временной составляющей. В этой базе хранится вся ключевая информация системы: пакеты модификаций для веб-сайтов, пользовательские настройки,

профили пользователей и другие важные данные проекта. Такая структура хранения позволяет эффективно работать с историческими данными и анализировать шаблоны использования системы с течением времени.

Для обмена информацией между компонентами системы разработаны соответствующие SQL-запросы и интерфейсы.

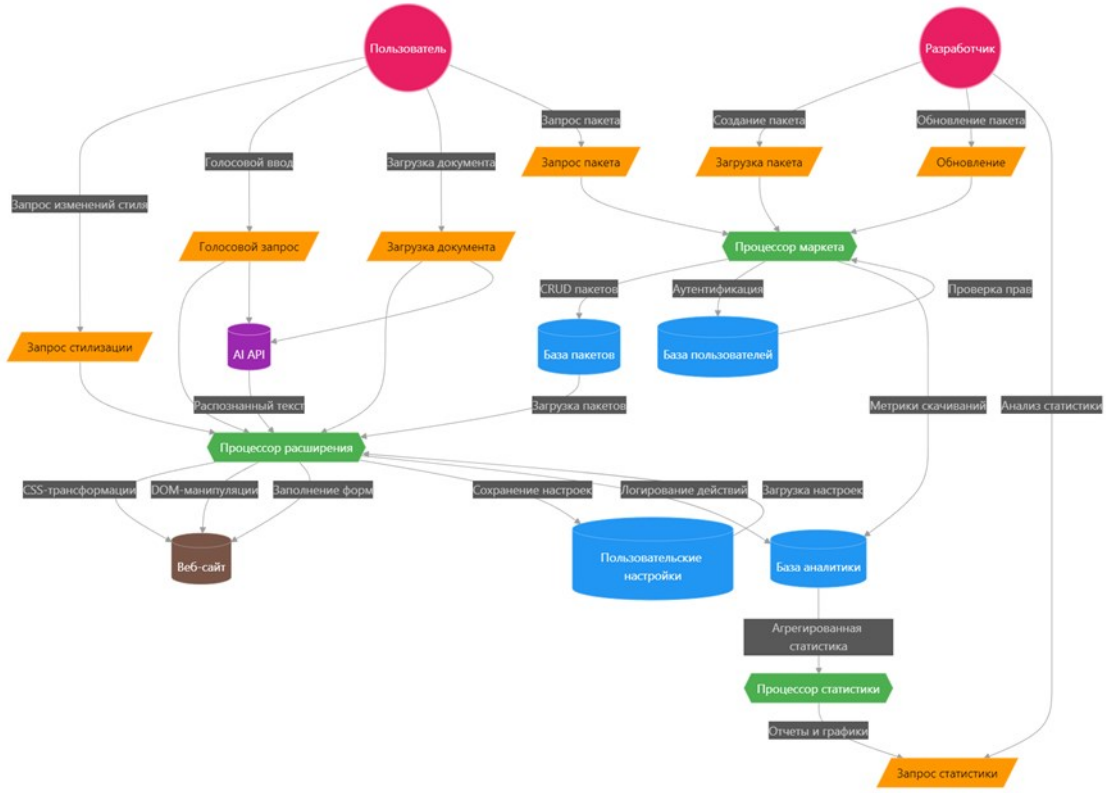


Рисунок 1. Последовательность обращений между компонентами системы

Многокомпонентная архитектура приложения позволяет создать надежное, масштабируемое и производительное расширение, способное обрабатывать большой объем информации о веб-страницах и применять к ним модификации для улучшения доступности. На рисунке 1 показана схема, отображающая последовательность взаимодействий между различными частями системы. Эта диаграмма наглядно демонстрирует порядок обмена данными и сигналами между компонентами приложения, позволяя увидеть, как именно выстроена коммуникация в процессе работы системы.

Диаграмма функций верхнего уровня представляет собой обобщённую схему, показывающую главные функциональные блоки системы и связи между ними. Она даёт общее представление о том, что делает система,

не вдаваясь в технические детали реализации или внутреннего устройства каждого компонента. Такая диаграмма помогает понять основное назначение системы и её ключевые возможности на концептуальном уровне.

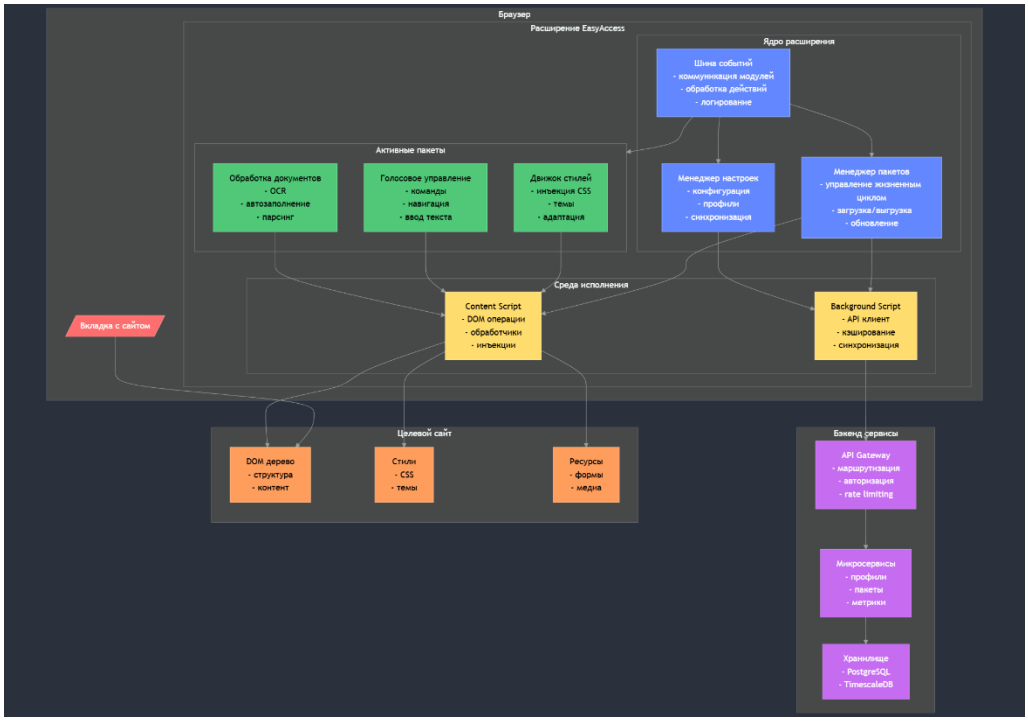


Рисунок 2. Функциональная диаграмма верхнего уровня

На рисунке 2 представлена обобщенная функциональная схема, иллюстрирующая основные рабочие процессы браузерного расширения EasyAccess. Центральным элементом архитектуры выступает подсистема адаптации веб-контента, которая обеспечивает повышение доступности просматриваемых пользователем страниц. Данная подсистема обрабатывает два основных типа входных данных: информацию о пользовательских предпочтениях и контекстные параметры веб-страниц. На основе анализа этих данных система применяет соответствующие модификации к отображаемым страницам, делая их более доступными для пользователей с различными потребностями.

Работа браузерного расширения EasyAccess осуществляется в соответствии с рядом нормативными актами и стандартов, таких как Web Content Accessibility Guidelines (WCAG) и ГОСТ Р 52872-2019. Все эти

факторы влияют на процесс преобразования веб-страниц и обеспечения доступности контента.

Процесс контролирует как разработчик и автор пакета модификаций, обеспечивающий соответствие модификаций требованиям и стандартам, так и конечный пользователь, имеющий возможность настраивать расширение под свои конкретные потребности.

PERT-диаграмма зависимостей участников команды используется для анализа и планирования процесса разработки, показывая взаимосвязь между различными участниками проекта и их зонами ответственности. Данная диаграмма помогает управлять проектом и обеспечивать эффективную коммуникацию и координацию между членами команды, что особенно важно в многокомпонентной системе с различными технологиями. PERT-диаграмма представлена на рисунке 3.

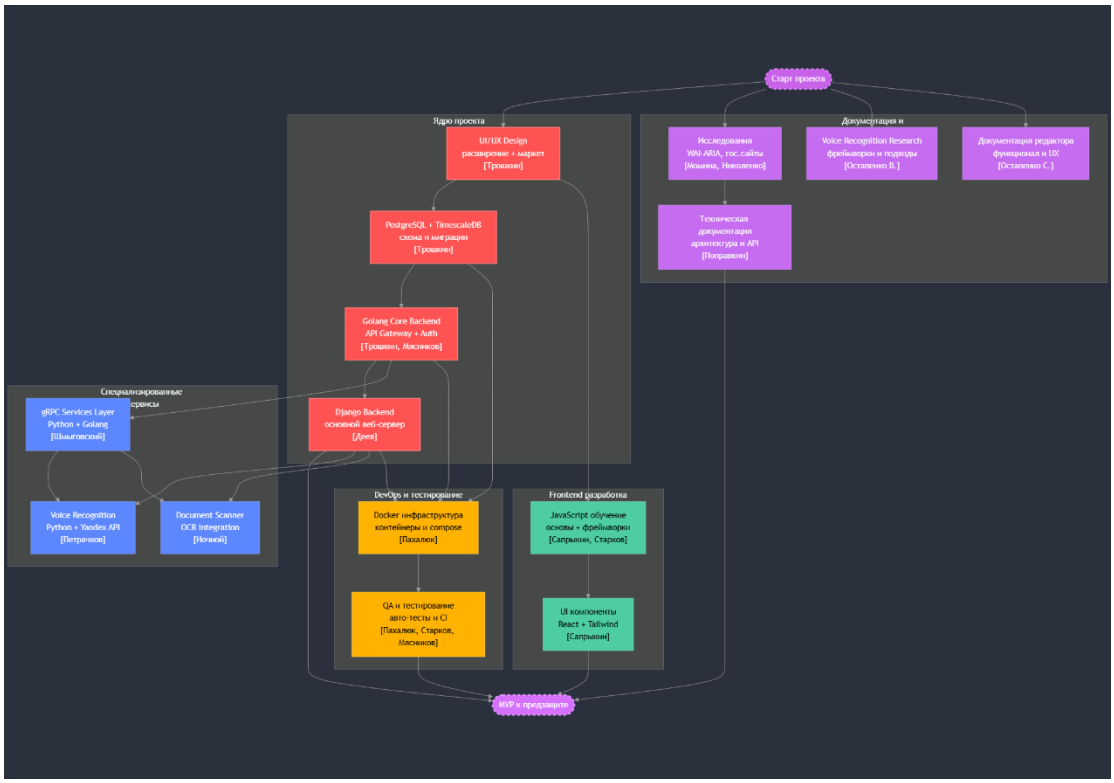


Рисунок 3. PERT-диаграмма зависимостей участников команды

Ниже, на рисунке 4, приведена диаграмма прецедентов, реализованных в программном приложении. Она является визуальным инструментом, который используется для описания функциональности системы с точки зрения ее взаимодействия с внешними пользователями (акторами) и другими

системами. Основная цель диаграммы прецедентов -- показать, как пользователи будут взаимодействовать с системой и какие функции они смогут использовать.

Рисунок 4. Диаграмма прецедентов

В проекте предусмотрено несколько типов пользователей:

- Основной пользователь (человек с ограниченными возможностями);
- Разработчик пакетов модификаций;
- Администратор маркетплейса.

Ниже приведены списки функций для каждого типа пользователя:

- **Пользователь:**
    1. Просмотр доступных пакетов: пользователи могут просматривать доступные пакеты модификаций в расширении;
    2. Управление пакетами: возможность включать и отключать пакеты модификаций по своему усмотрению;
    3. Импорт пакетов: возможность импортировать пакеты из файлов или через строковый код;
    4. Авторизация и регистрация в маркетплейсе: пользователи могут регистрироваться, чтобы получить доступ к маркетплейсу пакетов;
    5. Оценка пакетов: авторизованные пользователи имеют возможность оставлять рейтинги для пакетов модификаций;
    6. Установка пакетов из маркетплейса: возможность устанавливать пакеты из общего маркетплейса;
    7. Настройка личных параметров доступности: пользователь может настраивать базовые параметры доступности для всех сайтов.
- **Разработчик пакетов модификаций:**
    1. Создание пакетов: разработка новых пакетов модификаций для улучшения доступности;
    2. Редактирование пакетов: возможность обновлять и улучшать существующие пакеты;
    3. Тестирование пакетов: проверка работоспособности пакетов на различных веб-сайтах;
    4. Загрузка пакетов в маркетплейс: возможность поделиться своими разработками с сообществом;
    5. Получение отзывов: анализ оценок и отзывов пользователей для улучшения пакетов;
    6. Разработчик также может выполнять все функции, перечисленные для основного пользователя.
- **Администратор маркетплейса:**

1. Управление пакетами модификаций: возможность одобрять, отклонять и удалять пакеты из маркетплейса;
2. Управление пользователями: возможность модерировать пользователей и их действия;
3. Просмотр статистики: доступ к статистике использования и популярности пакетов;
4. Управление категориями: организация пакетов по категориям для удобства пользователей;
5. Администратор также имеет доступ ко всем функциям разработчика и основного пользователя.

2.2 Проектирование базы данных (Деев Е., Трошкин Д.)

База данных является основой любого программного приложения, тем более направленного на обеспечение доступности веб-сайтов. Для проекта EasyAccess база данных играет решающую роль в хранении и организации информации о пакетах модификаций, пользователях, настройках и других сущностях. Она обеспечивает сохранение данных, необходимых для функционирования расширения, что позволяет приложению эффективно обрабатывать и предоставлять информацию в ответ на запросы пользователей. Проектирование базы данных началось с исследования предметной области, выявления основных объектов-сущностей, процессов и связей между этими элементами. Затем были определены атрибуты каждого объекта-сущности и каждого процесса, назначены первичные ключи, построена ER-диаграмма как модель предметной области. На базе ER-диаграммы была создана реляционная база данных, которая приведена к 3-й нормальной форме с использованием технологии нормализации. На очередном этапе проекта была осуществлена имплементация слоя персистентности посредством развертывания базы данных PostgreSQL с интегрированным расширением TimescaleDB. Выбор данного технологического стека был обусловлен необходимостью эффективной обработки временных рядов, являющихся ключевым типом

данных в системе мониторинга и анализа пользовательской активности. Такая архитектура хранилища позволяет оптимально масштабировать инфраструктуру с учетом специфических паттернов обращения к разнотипным темпоральным данным, используемым в операциях статистического анализа и формирования отчетов.

ER-диаграмма представляет собой графическое отображение модели данных, которое используется для описания отношений между сущностями в предметной области. Она позволяет наглядно представить структуру базы данных и связи между ее компонентами, что облегчает понимание и работу с данными.

ER-диаграмма в нотации Баркера приведена на рисунке 5. Данная диаграмма соответствует функционалу браузерного расширения EasyAccess, включая управление пакетами модификаций, пользователями, настройками и другими сущностями.

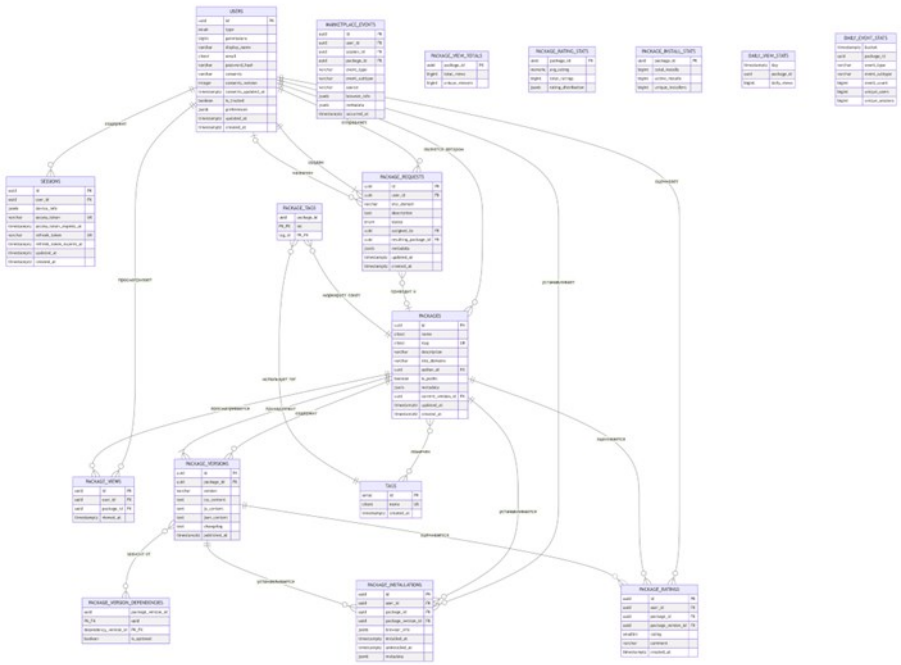


Рисунок 5. ER-диаграмма базы данных

На ER-диаграмме представлены основные сущности системы и их взаимосвязи:

1. **Users (Пользователи)** - хранит информацию о пользователях системы, включая их идентификаторы, имена, email и настройки.
2. **Sessions (Сессии)** - отслеживает сессии пользователей для обеспечения безопасности и аутентификации.
3. **Packages (Пакеты)** - содержит информацию о пакетах модификаций, включая название, описание, домены применения и другие метаданные.
4. **Package Versions (Версии пакетов)** - хранит различные версии пакетов модификаций, включая код CSS и JavaScript.
5. **Tags (Теги)** - категории и метки для пакетов, позволяющие группировать и фильтровать их.
6. **Package Requests (Запросы на создание пакетов)** - запросы от пользователей на разработку новых пакетов.
7. **Package Views (Просмотры пакетов)** - статистика просмотров пакетов для анализа популярности.
8. **Package Installations (Установки пакетов)** - данные об установке пакетов пользователями.
9. **Package Ratings (Рейтинги пакетов)** - оценки и отзывы пользователей о пакетах.
10. **Marketplace Events (События маркетплейса)** - различные события, происходящие на маркетплейсе, для анализа и аудита.

Каждая сущность имеет определенный набор атрибутов, которые представляют собой характеристики или свойства этой сущности. Например, сущность Users имеет атрибуты id, email, display_name, password_hash и другие, которые хранят информацию о пользователе.

Связи между сущностями показывают, как они взаимодействуют друг с другом. Например, связь между сущностями Users и Packages показывает, что пользователь может создавать пакеты модификаций, а связь между Packages и Package Versions указывает на то, что пакет может иметь несколько версий.

Эта ER-диаграмма является основой для создания реляционной базы данных, которая реализует всю функциональность системы и обеспечивает эффективное хранение и управление данными.

Выбор системы управления базами данных

В качестве основной СУБД для проекта был выбран PostgreSQL с расширением TimescaleDB. Данный выбор обусловлен следующими факторами:

1. **Надежность и стабильность** – PostgreSQL имеет репутацию надежной и проверенной временем СУБД, способной обеспечить стабильную работу под высокой нагрузкой.
2. **Поддержка JSON/JSONB типов данных** – позволяет гибко хранить структурированные данные, такие как настройки пакетов модификаций и пользовательские предпочтения.
3. **Расширяемость** – возможность подключения дополнительных модулей и расширений, в том числе TimescaleDB для эффективной работы с временными рядами данных.
4. **Поддержка сложных запросов** – развитые возможности SQL, включая оконные функции, рекурсивные запросы и другие продвинутые возможности.
5. **Масштабируемость** – возможность горизонтального и вертикального масштабирования для обеспечения роста проекта.

Концептуальное проектирование

На этапе концептуального проектирования были выделены основные сущности системы и их взаимосвязи:

1. **Users (Пользователи)** — информация о пользователях маркетплейса.
2. **Sessions (Сессии)** — данные о сессиях пользователей.
3. **Packages (Пакеты)** — наборы модификаций для веб-сайтов.
4. **Package Versions (Версии пакетов)** — различные версии пакетов модификаций.

5. **Tags (Теги)** — категории и метки для пакетов.
6. **Package Requests (Запросы на создание пакетов)** — запросы от пользователей на разработку новых пакетов.
7. **Package Views (Просмотры пакетов)** — статистика просмотров пакетов.
8. **Package Installations (Установки пакетов)** — данные об установке пакетов пользователями.
9. **Package Ratings (Рейтинги пакетов)** — оценки и отзывы пользователей о пакетах.
10. **Marketplace Events (События маркетплейса)** — различные события, происходящие на маркетплейсе.

Логическое проектирование

На этапе логического проектирования были определены атрибуты каждой сущности, их типы данных, первичные и внешние ключи, а также ограничения целостности. Основные таблицы и их структура представлены на рисунке 5.

Схема включает следующие типы связей:

1. **Один-ко-многим** (1:N):
   - Пользователь может создать множество пакетов;
   - Пользователь может иметь множество сессий;
   - Пакет может иметь множество версий;
   - Пользователь может оставить множество оценок;
   - Пакет может иметь множество просмотров;
   - Пакет может быть установлен множество раз.
2. **Многие-ко-многим** (N:M):
   - Пакеты могут иметь множество тегов, а теги могут относиться к множеству пакетов;
   - Версии пакетов могут зависеть от других версий пакетов.
3. **Один-к-одному** (1:1):

- Пакет имеет одну текущую версию;
- Запрос на создание пакета может быть связан с одним результирующим пакетом.

Физическое проектирование

На этапе физического проектирования были определены детали хранения данных, индексы, триггеры и другие механизмы оптимизации:

1. **Индексы** — созданы индексы для ускорения поиска и фильтрации:
   - индексы на внешних ключах (user_id, package_id и т. д.),
   - текстовые индексы для полнотекстового поиска (name, description),
   - индексы для часто используемых условий фильтрации (is_public, created_at).
2. **Гипертаблицы (TimescaleDB)** — для хранения временных рядов данных:
   - package_views — для отслеживания просмотров,
   - marketplace_events — для отслеживания событий,
   - package_installations — для отслеживания установок,
   - package_ratings — для отслеживания оценок.
3. **Материализованные представления** – для оптимизации часто выполняемых агрегирующих запросов:
   - package_view_totals — агрегация статистики просмотров,
   - package_total_stats — агрегация статистики событий,
   - package_install_stats — агрегация статистики установок,
   - package_rating_stats — агрегация статистики оценок.
4. **Триггеры** — для автоматического обновления данных:
   - автоматическое обновление current_version_id при добавлении новой версии,
   - проверка циклических зависимостей между версиями пакетов,

- генерация slug из названия пакета при создании.

5. **Задания по расписанию** — для периодического обновления материализованных представлений:
    - refresh_marketplace_stats — периодическое обновление статистики.

2.3 Реализация базы данных (Деев Е., Шмыговский Н., Трошкин Д., Мясников Д.)

Для реализации базы данных был выбран PostgreSQL с расширением TimescaleDB из-за его широких возможностей, высокой производительности и поддержки для работы с временными рядами данных. Эта система управления базами данных обеспечивает надежное хранение данных, эффективное выполнение запросов и поддержку множества функций, необходимых для реализации проекта EasyAccess.

В процессе реализации базы данных, придерживаясь логической структуре, представленной на рисунке 5, была создана база данных с соответствующей схемой, отражающей структуру всех таблиц, типы полей и другие атрибуты. Каждая таблица была создана с учетом ее логической модели, определяя не только названия таблиц и полей, но и их типы данных, ограничения, индексы и так далее.

Особое внимание в проекте уделяется работе с временными данными, такими как просмотры пакетов, установки и рейтинги. Для этого используется расширение TimescaleDB, которое позволяет эффективно хранить и запрашивать большие объемы временных данных. Это обеспечивает высокую производительность при работе с аналитикой и статистикой использования пакетов модификаций.

В разработке базы данных используются стандартные запросы SQL, а также специализированные запросы для работы с TimescaleDB, специально адаптированные для поддержки основной деятельности расширения. Каждый запрос заранее проработан, протестирован и оптимизирован для

бесперебойной работы на сервере, обеспечивая эффективную функциональность всей системы.

Для обеспечения высокой производительности и масштабируемости были разработаны миграции, которые позволяют последовательно применять изменения и поддерживать согласованное состояние схемы базы данных. В листинге 1 представлен пример кода миграции для инициализации базовых расширений и создания таблицы пользователей.

Листинг 1. Инициализация базовых расширений и создание таблицы пользователей

```
create extension if not exists citext;
create extension if not exists pg_trgm;
create extension if not exists "uuid-ossp";

create type user_type as enum ('anonymous', 'registered');

create table users (
        id uuid primary key default uuid_generate_v4(),
        type user_type default 'anonymous',
        permissions bigint default 0,
        display_name varchar(128),
        email citext unique
                check (length(email) between 5 and 255)
                check (email ~ '^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,}$'),
        password_hash varchar(255),
        consents varchar[] default '{}',
        consents_version integer default 1,
        consents_updated_at timestamptz,
        is_trusted boolean default false,
        preferences jsonb default '{}',
        updated_at timestamptz default now(),
        created_at timestamptz default now()
);
```

Для эффективной работы с базой данных был разработан слой абстракции, включающий репозитории для работы с различными сущностями. В листинге 2 представлен пример кода для подключения к базе данных.

Листинг 2. Подключение к базе данных

```
type Database struct {
	db *ksql.DB
	pool *pgxpool.Pool
	logger *logger.Logger
	config *config.DatabaseConfig
}

func New(ctx context.Context, cfg *config.DatabaseConfig, log *logger.Logger) (*Database) {
	log.Info().Msg("connecting to database")
	poolConfig, err := pgxpool.ParseConfig(cfg.URL)
	if err != nil {
		return nil, errors.Wrap(err, "parse connection string")
	}

	poolConfig.MaxConns = int32(cfg.MaxOpenConns)
	poolConfig.MinConns = int32(cfg.MaxIdleConns)
	poolConfig.MaxConnLifetime = cfg.ConnMaxLifetime
	poolConfig.MaxConnIdleTime = cfg.ConnMaxIdleTime
	poolConfig.ConnConfig.ConnectTimeout = 5 * time.Second

	pool, err := pgxpool.NewWithConfig(ctx, poolConfig)
	if err != nil {
		return nil, errors.Wrap(err, "connect to database")
	}

	if err = pool.Ping(ctx); err != nil {
		return nil, errors.Wrap(err, "ping database")
	}
```

```
db, err := kpgx.NewFromPgxPool(pool)
 if err != nil {
        return nil, errors.Wrap(err, "create ksql adapter")
}
log.Info().
        Int("max_conns", cfg.MaxOpenConns).
        Int("min_conns", cfg.MaxIdleConns).
        Msg("database connected")
return &Database{ db: &db, pool: pool, logger: log, config: cfg, }, nil }
```

Для оптимизации запросов и повышения производительности были созданы материализованные представления, которые позволяют быстро получать статистику о просмотрах, установках и рейтингах пакетов. В листинге 3 представлен пример кода для создания материализованного представления для статистики просмотров.

Листинг 3. Создание материализованного представления

```
create materialized view package_view_totals as
select package_id,
        count(*) as total_views,
        count(distinct user_id) as unique_viewers
from package_views
group by package_id with no data;
create unique index idx_view_totals_package on package_view_totals (package_id);
```

Для работы с временными рядами данных используются гипертаблицы TimescaleDB, которые обеспечивают эффективное хранение и запрос временных данных. В листинге 4 представлен пример кода настройки гипертаблиц.

Листинг 4. Настройка гипертаблиц TimescaleDB

```
select create_hypertable('package_views', 'viewed_at');
select create_hypertable('marketplace_events', 'occurred_at');
select create_hypertable('package_installations', 'installed_at');
```

Для обеспечения удобства использования и повышения производительности были также разработаны автоматические задачи для периодического обновления материализованных представлений. В листинге 5 представлен пример кода для настройки задач.

Листинг 5. Настройка задач для автоматического обновления статистики

```
CREATE OR REPLACE PROCEDURE refresh_marketplace_stats(job_id int, config jsonb)
LANGUAGE PLPGSQL AS
$$
BEGIN
        BEGIN
                REFRESH MATERIALIZED VIEW CONCURRENTLY package_view_totals;
                RAISE NOTICE 'package_view_totals updated';
        EXCEPTION
                WHEN OTHERS THEN
                        RAISE WARNING 'package_view_totals error: %', SQLERRM;
        END;
        ...
END;
$$;

SELECT add_job('refresh_marketplace_stats','1h',
        initial_start => now(),
        fixed_schedule => true,
        timezone => 'UTC'
);
```

Интеграция с Supabase, современной платформой для разработки, используется для обеспечения безопасного управления пользователями и аутентификации. Это позволяет эффективно управлять пользователями, сессиями и правами доступа без необходимости разрабатывать эти системы с нуля. Supabase предоставляет надежную и безопасную инфраструктуру для аутентификации, управления базой данных и хранения файлов.

Реализованная база данных для проекта EasyAccess полностью соответствует поставленным требованиям и обеспечивает:

1. Эффективное хранение данных - структура таблиц оптимизирована для хранения всех необходимых данных с минимальной избыточностью.

2. Высокую производительность - применение индексов, материализованных представлений и других оптимизаций обеспечивает быструю обработку запросов.

3. Масштабируемость - использование TimescaleDB для работы с временными рядами данных позволяет эффективно масштабировать систему при росте количества пользователей.

4. Целостность данных - система ограничений, триггеров и проверок обеспечивает целостность и согласованность данных.

5. Гибкость - использование JSONB полей позволяет хранить сложные структурированные данные без необходимости изменения схемы базы данных.

База данных является надежным фундаментом для всего проекта, обеспечивая хранение и обработку данных для всех компонентов системы: от управления пакетами модификаций до отслеживания пользовательской активности и формирования статистики.

2.4 Разработка программного обеспечения (Трошкин Д., Сапрыкин П., Петрачков В.)

Разработка программного обеспечения для браузерного расширения EasyAccess, основанного на современных технологиях, требует комплексного подхода к интерфейсу пользователя и логике обработки данных для обеспечения высокого качества и производительности. Важно, чтобы дизайн интерфейса был интуитивно понятным и привлекательным для пользователей, а серверная часть обеспечивала надежность, безопасность и эффективную работу приложения.

Дизайн браузерного расширения разработан с использованием языков CSS, HTML, TypeScript и JavaScript. Эти языки позволяют создавать структуру страниц, стилизировать элементы интерфейса и добавлять интерактивные функции, обеспечивая высокий уровень пользовательского опыта. Основные моменты, которые будут рассмотрены далее, включают палитру цветов, структуру интерфейса и компонентную архитектуру. Структура расширения была продумана с учетом лучших практик в области разработки расширений для браузеров, все элементы, их внешний вид и расположение были задуманы с учетом удобства использования и эстетической привлекательности. Это позволяет пользователям получить удовлетворение от взаимодействия с расширением, легко находить необходимую информацию и выполнять нужные действия.

При разработке палитры цветов были выбраны основные цвета #2563eb (синий) и #E6ECFA (светло-голубой) для акцентных элементов, а также нейтральные цвета для обеспечения хорошей читаемости и контрастности интерфейса. Эти цвета были подобраны с учетом эстетической гармонии, удобства восприятия и функциональности интерфейса.

Структура расширения включает несколько основных экранов:

1. Главный экран - отображает информацию о текущей вкладке и установленных пакетах модификаций

2. Экран просмотра пакета - предоставляет детальную информацию о выбранном пакете

3. Редактор пакетов - позволяет создавать и редактировать пакеты модификаций

4. Маркетплейс - предоставляет доступ к общему хранилищу пакетов модификаций

Расширение было разработано с использованием компонентного подхода, где каждый элемент интерфейса представляет собой отдельный компонент с собственной логикой и стилями. Это позволяет повысить

переиспользуемость кода, улучшить его поддерживаемость и обеспечить единообразие интерфейса.

Для реализации клиентской части расширения был выбран фреймворк Solid.js, который предоставляет высокую производительность и реактивность при малом размере библиотеки. Это особенно важно для браузерных расширений, где размер кода напрямую влияет на скорость загрузки и отзывчивость интерфейса.

Компонент Logo отвечает за отображение логотипа расширения и его названия. Он использует SVG для отображения иконки и имеет возможность отображать компактную версию без текста. Такой подход позволяет гибко использовать компонент в различных частях интерфейса.

Компонент HomePage является главным экраном расширения и объединяет в себе несколько других компонентов: TabInfo для отображения информации о текущей вкладке, DevModeWarning для предупреждения о необходимости включения режима разработчика в браузере и InstalledPackages для отображения списка установленных пакетов модификаций.

В расширении используется компонентный подход, где каждый элемент интерфейса представляет собой отдельный компонент с собственной логикой и стилями. Это позволяет повысить переиспользуемость кода, улучшить его поддерживаемость и обеспечить единообразие интерфейса.

Для управления состоянием приложения используется библиотека Pinia, которая предоставляет реактивное и гибкое хранилище состояния.

Компонент Button представляет собой многофункциональную кнопку с возможностью отображения выпадающего меню. Компонент поддерживает различные стили, иконки и выравнивание, что делает его универсальным инструментом для создания интерфейса. Выпадающее меню позволяет предоставить пользователю дополнительные опции без загромождения интерфейса.

Модуль управления редактором пакетов является центральным элементом для создания и редактирования пакетов модификаций. Он

управляет состоянием редактора, включая метаданные пакета (название, описание, иконка, URL), содержимое пакета (CSS и JavaScript код), состояние интерфейса пользователя и процесс сохранения пакета. Модуль использует паттерн "хранилище состояния" (store), который обеспечивает централизованное управление состоянием приложения и реактивное обновление интерфейса пользователя.

Модуль для работы с хранилищем пакетов обеспечивает взаимодействие с локальной базой данных IndexedDB через библиотеку Dexie.js. Он предоставляет функции для получения, добавления, обновления и удаления пакетов модификаций. Особое внимание уделяется функции getScriptsForDomain, которая отвечает за выбор пакетов, применимых к конкретному домену, что является ключевой функциональностью расширения. Для улучшения пользовательского опыта и обеспечения интерактивности интерфейса, в приложении используются хуки React, которые инкапсулируют логику работы с данными и предоставляют удобный интерфейс для компонентов.

Хуки для работы с базой данных предоставляют удобный интерфейс для компонентов, позволяя им получать и отслеживать изменения данных в базе. Функции createLiveValue и createLiveQuery являются обертками над библиотекой Dexie, которые обеспечивают реактивность данных в интерфейсе пользователя. Хуки useScripts, useScript и useScriptsForDomain инкапсулируют логику работы с конкретными запросами к базе данных, предоставляя компонентам простой и понятный интерфейс.

Модуль инъекции скриптов является одним из ключевых компонентов расширения, отвечающих за внедрение пользовательских CSS и JavaScript кода в веб-страницы. Он обеспечивает регистрацию скриптов в браузере, проверку соответствия доменов и внедрение стилей CSS в активные вкладки. Особое внимание уделяется безопасности, с проверкой URL-адресов и исключением системных страниц браузера, а также подробным логированием процесса инъекции для отладки и мониторинга.

Для маркетплейса пакетов модификаций был разработан специальный модуль для взаимодействия с Supabase, который обеспечивает аутентификацию пользователей и управление пакетами.

Функции для работы с маркетплейсом обеспечивается взаимодействие с серверной частью приложения через Supabase. Функция fetchScripts отвечает за загрузку пакетов из маркетплейса с поддержкой поиска, сортировки и пагинации. Функция installScript обеспечивает установку пакета из маркетплейса в локальное хранилище пользователя, а также обновление статистики загрузок на сервере.

Суммируя проведенную работу, можно выделить следующие ключевые особенности разработанного программного обеспечения:

1. Компонентная архитектура с использованием Solid.js обеспечивает модульность, переиспользуемость и высокую производительность.
2. Хранилище состояния на основе Pinia обеспечивает централизованное управление состоянием приложения и реактивное обновление интерфейса пользователя.
3. Локальная база данных на основе IndexedDB через Dexie.js обеспечивает эффективное хранение и управление данными на стороне клиента.
4. Интеграция с Supabase обеспечивает безопасную аутентификацию и управление данными на стороне сервера.
5. Механизмы инъекции скриптов обеспечивают внедрение пользовательских CSS и JavaScript кода в веб-страницы с учетом безопасности и производительности.

Разработанное программное обеспечение обеспечивает полнофункциональное браузерное расширение для повышения доступности веб-сайтов, предоставляющее пользователю широкие возможности для настройки и улучшения доступности веб-страниц.

2.5 Система безопасности (Шмыговский Н., Трошкин Д.)

В современном мире, где обеспечение доступности веб-ресурсов становится все более важным, безопасность браузерных расширений, таких как EasyAccess, играет ключевую роль. Разработка расширения, которое модифицирует содержимое веб-страниц, требует особого внимания к безопасности, чтобы защитить как пользователей, так и веб-сайты.

При создании и обслуживании системы безопасности браузерного расширения EasyAccess важно обеспечить защиту конфиденциальности пользователей и предотвратить несанкционированный доступ к пользовательским данным. В этом контексте система безопасности включает в себя меры при регистрации, авторизации и хранении пользовательских настроек.

Для защиты пользовательских данных в маркетплейсе пакетов применяется авторизация и аутентификация через Supabase, который обеспечивает безопасное хранение и проверку пользовательских учетных данных. При регистрации пароль пользователя хешируется и безопасно хранится на сервере, что предотвращает возможность получения исходного пароля даже в случае компрометации базы данных.

Функции регистрации и авторизации пользователей обеспечивают безопасное управление учетными данными через Supabase. При регистрации создается новый пользователь, и его данные безопасно хранятся на сервере. При авторизации проверяется соответствие введенных данных с данными, хранящимися на сервере, и в случае успеха пользователь получает доступ к своей учетной записи.

Особое внимание уделяется безопасности при работе с анонимными пользователями. Для них создаются временные идентификаторы, которые позволяют отслеживать их активность без необходимости регистрации, но при этом обеспечивая необходимый уровень безопасности.

Для обеспечения безопасности пользовательских данных и предотвращения несанкционированного доступа к ним, в расширении используются следующие механизмы:

1. Разделение прав доступа - обычные пользователи не могут управлять чужими пакетами модификаций или получать доступ к конфиденциальным данным других пользователей.

2. Валидация входных данных - все пользовательские данные проходят проверку на корректность и безопасность перед сохранением в базе данных.

3. Безопасное хранение пользовательских предпочтений - настройки пользователя хранятся в локальном хранилище и доступны только самому пользователю.

4. Контроль доступа к API - доступ к API-серверу контролируется с помощью токенов доступа, которые проверяются на каждый запрос.

Особое внимание уделяется безопасности при инъекции скриптов в веб-страницы. Расширение проверяет доменные имена и URL-адреса перед применением модификаций, исключает системные страницы браузера и обеспечивает изоляцию скриптов для предотвращения конфликтов с основным кодом страницы.

Кроме того, для обеспечения безопасности пользовательских данных при обмене пакетами модификаций используется механизм кодирования и декодирования данных, который предотвращает несанкционированное изменение или перехват данных.

Для обеспечения безопасности при хранении данных в браузере используется механизм индексированной базы данных (IndexedDB), который обеспечивает изоляцию данных между различными расширениями и веб-сайтами. Это предотвращает несанкционированный доступ к данным расширения со стороны других веб-приложений или расширений.

Комплексный подход к системе безопасности в браузерном расширении EasyAccess обеспечивает защиту личных данных пользователей и создает доверие к безопасности использования расширения. Эти меры способствуют

обеспечению уверенности пользователей в защите их личной информации, что играет важную роль в поддержании успеха и репутации расширения.

2.6 Тестирование приложения (Момина А., Николенко Д., Старков Р., Пахалюк И., Деев Е.)

Отладка и тестирование браузерного расширения EasyAccess являются важными этапами разработки, и в этом параграфе будут рассмотрены основные методы и инструменты, используемые для их проведения. Основной целью является выявление и устранение ошибок, а также проверка функциональности и соответствия требованиям. Это позволит улучшить качество работы расширения и повысить уровень удовлетворенности пользователей.

Тестирование проводилось с использованием нескольких методов. Применялось ручное тестирование, которое охватывало все ключевые функции и сценарии использования расширения, выявляя ошибки в интерфейсе и логике работы. Кроме того, использовался анализ логов приложения для выявления и устранения скрытых проблем, которые могли бы остаться незамеченными при других методах тестирования. В таблице 1 представлены результаты отладки и тестирования программы.

Таблица 1. Результаты отладки и тестирования программы

№ теста	**Входные данные**	**Вводимое значение**	**Ожидаемая реакция программы**	**Фактическая реакция программы**	**Ошибка выявлена**
1	-	-	Загрузка главного экрана расширения успешна	Главный экран загружен успешно	Нет
2	Нажатие на кнопку со стрелкой в списке пакетов	-	Переход на страницу детального просмотра пакета	Переход выполнен успешно	Нет
3	Нажатие на кнопку "Импорт"	-	Переход на страницу импорта пакета	Переход выполнен успешно	Нет

4	Поле ввода для импорта пакета	Некорректный код	Отображение сообщения об ошибке	Отображено сообщение "Неверный формат пакета"	Нет
5	Поле ввода для импорта пакета	Корректный код пакета	Успешный импорт пакета и переход на страницу пакета	Импорт выполнен успешно, переход на страницу пакета	Нет
6	Нажатие на кнопку переключения пакета	-	Изменение состояния пакета (включен/выключен)	Состояние пакета изменено успешно	Нет
7	Нажатие на кнопку "Редактировать"	-	Открытие редактора пакета	Редактор открыт успешно	Нет
8	Нажатие на кнопку "Поделиться"	-	Копирование кода пакета в буфер обмена	Код скопирован успешно	Нет
9	Нажатие на пункт "Экспорт в файл" в выпадающем меню кнопки "Поделиться"	-	Сохранение пакета в файл	Файл сохранен успешно	Нет
10	Нажатие на кнопку "Удалить"	-	Отображение кнопки подтверждения удаления	Кнопка подтверждения отображена	Нет
11	Нажатие на кнопку подтверждения удаления	-	Удаление пакета и возврат на главный экран	Пакет удален, возврат на главный экран выполнен	Нет
12	Нажатие на кнопку "Создать"	-	Открытие редактора для создания нового пакета	Редактор открыт успешно	Нет
13	Поля редактора пакета	Название: пустое, Остальные поля: заполнены	Отображение предупреждения о необходимости указать название	Отображено предупреждение "Укажите название пакета!"	Нет
14	Поля редактора пакета	Все поля заполнены корректно	Сохранение пакета и отображение уведомления об успешном сохранении	Пакет сохранен, уведомление отображено	Нет
15	Смена активной вкладки в браузере	-	Обновление информации о текущей вкладке в расширении	Информация обновлена успешно	Нет

16	Открытие маркетплейса	-	Загрузка списка доступных пакетов	Список загружен успешно	Нет
17	Поле поиска в маркетплейсе	"тест"	Отображение результатов поиска по запросу	Результаты поиска отображены	Нет
18	Выбор сортировки в маркетплейсе	Сортировка по рейтингу	Изменение порядка отображения пакетов	Порядок отображения изменен	Нет
19	Нажатие на кнопку "Установить" в карточке пакета в маркетплейсе	-	Установка пакета и отображение сообщения об успешной установке	Пакет установлен, сообщение отображено	Нет
20	Нажатие на кнопку "Войти" в маркетплейсе	-	Открытие модального окна авторизации	Модальное окно открыто	Нет
21	Поля авторизации	Email: некорректный формат, Пароль: корректный	Отображение сообщения об ошибке ввода email	Сообщение об ошибке отображено	Нет
22	Поля авторизации	Email: корректный, Пароль: слишком короткий	Отображение сообщения об ошибке ввода пароля	Сообщение об ошибке отображено	Нет
23	Поля авторизации	Email: корректный, Пароль: корректный, Неверные данные	Отображение сообщения о неверных учетных данных	Сообщение об ошибке отображено	Нет
24	Поля авторизации	Email: корректный, Пароль: корректный, Верные данные	Успешная авторизация и обновление интерфейса	Авторизация выполнена успешно, интерфейс обновлен	Нет
25	Нажатие на кнопку "Загрузить" для пользователя с пакетами	-	Отображение списка доступных для загрузки пакетов	Список отображен корректно	Нет
26	Выбор пакета для загрузки в маркетплейс	-	Загрузка пакета в маркетплейс и отображение сообщения об успешной загрузке	Пакет загружен успешно, сообщение отображено	Нет
27	Нажатие на карточку пакета в маркетплейсе,	-	Кнопка "Установить" неактивна или	Кнопка неактивна, отображен	Нет

	уже установленного пользователем		отображает статус "Установлен"	статус "Установлен"	
28	Оценка пакета в маркетплейсе	Выбор рейтинга (звездочки)	Сохранение оценки и обновление рейтинга пакета	Оценка сохранена, рейтинг обновлен	Нет
29	Нажатие на кнопку "Выйти" в маркетплейсе	-	Выход из учетной записи и обновление интерфейса	Выход выполнен успешно, интерфейс обновлен	Нет
30	Переключение между вкладками CSS и JavaScript в редакторе	-	Изменение отображаемого содержимого редактора	Содержимое изменено успешно	Нет
31	Смена темы редактора	-	Изменение цветовой схемы редактора	Цветовая схема изменена успешно	Нет
32	Нажатие на кнопку "Перейти в корзину"	-	Переход на страницу "Корзина"	Переход выполнен успешно	Нет
33	Нажатие на кнопку "Обновить" в редакторе при редактировании существующего пакета	-	Сохранение изменений и отображение уведомления об успешном обновлении	Изменения сохранены, уведомление отображено	Нет

Тестирование расширения на различных сайтах показало его эффективность в повышении доступности веб-контента. Особенно положительные результаты были получены при использовании предустановленных пакетов модификаций для популярных сайтов, которые существенно улучшали их восприятие и использование людьми с различными ограничениями.

Для анализа пользовательского опыта было проведено тестирование с участием потенциальных пользователей, в том числе людей с различными видами ограничений. Результаты этого тестирования позволили выявить и устранить неочевидные проблемы и улучшить общее качество расширения.

В процессе тестирования также были выявлены потенциальные направления для дальнейшего развития расширения, такие как добавление новых функций для работы с аудио-контентом, улучшение распознавания элементов страницы и расширение библиотеки предустановленных пакетов модификаций.

2.7 Руководство по использованию программы

Руководство системного программиста

Назначение программы

Программа предназначена для установки, настройки и разработки браузерного расширения EasyAccess, обеспечивающего повышение доступности веб-сайтов для людей с ограниченными возможностями здоровья.

Технические и программные средства

- Visual Studio Code или другая IDE для разработки
- Node.js (версия 16.0.0 или выше)
- WXT (Web Extension Tools) для сборки и тестирования расширения
- PostgreSQL с расширением TimescaleDB для серверной части
- Браузер с поддержкой WebExtensions API (Chrome, Edge, Firefox)

Установка и настройка среды разработки

1. Установка Node.js:
    - Скачайте и установите Node.js с официального сайта: https://nodejs.org/
    - Проверьте установку с помощью команды: node -v
2. Клонирование репозитория:
    - Выполните команду: git clone https://github.com/easyaccess-team/extension.git
    - Перейдите в директорию проекта: cd extension
3. Установка зависимостей:
    - Выполните команду: npm install

4. Настройка базы данных (для серверной части):
    - Установите PostgreSQL: https://www.postgresql.org/download/
    - Установите расширение TimescaleDB: https://docs.timescale.com/install/
    - Создайте базу данных и пользователя согласно конфигурации в файле src/lib/storage/supabase.ts
5. Настройка Supabase (для маркетплейса):
    - Зарегистрируйтесь на Supabase: https://supabase.io/
    - Создайте новый проект
    - Настройте аутентификацию и таблицы согласно схеме базы данных из документации
    - Обновите данные для подключения в файле .env

Структура проекта

- src/ - исходный код расширения
    - components/ - компоненты интерфейса
    - entrypoints/ - точки входа для расширения
    - hooks/ - хуки для работы с данными
    - lib/ - вспомогательные функции и утилиты
    - models/ - модели данных
    - stores/ - хранилища состояния
    - styles/ - стили приложения
- public/ - статические файлы
- prisma/ - схема базы данных и миграции

Сборка и запуск

1. Запуск в режиме разработки:
    - Выполните команду: npm run dev
    - Расширение будет собрано и доступно для загрузки в браузер
2. Сборка для продакшена:

- Выполните команду: npm run build
- Расширение будет собрано в директории .output

3. Создание архива:
    - Выполните команду: npm run zip
    - Будет создан архив с расширением для публикации

Установка расширения в браузер

1. Откройте страницу управления расширениями в браузере
2. Включите режим разработчика
3. Нажмите "Загрузить распакованное расширение"
4. Выберите директорию .output из сборки расширения

Руководство пользователя

Назначение программы

Браузерное расширение EasyAccess предназначено для повышения доступности веб-сайтов для людей с ограниченными возможностями здоровья. Расширение позволяет модифицировать внешний вид и функциональность веб-страниц, делая их более удобными для использования различными категориями пользователей.

Основные функции программы

- Установка и управление пакетами модификаций для веб-сайтов
- Настройка параметров отображения веб-страниц (размер текста, контрастность и т.д.)
- Импорт и экспорт пакетов модификаций
- Доступ к маркетплейсу пакетов модификаций
- Создание и редактирование собственных пакетов модификаций

Системные требования

- Операционная система: Windows, macOS, Linux
- Браузер: Chrome (версия 88+), Edge (версия 88+), Firefox (версия 78+)

- Включенный режим разработчика в браузере

Установка расширения

1. Скачайте расширение из Chrome Web Store, Mozilla Add-ons или напрямую с сайта проекта
2. Установите расширение в браузер согласно инструкциям браузера
3. После установки на панели инструментов появится иконка расширения EasyAccess

Использование расширения

1. Нажмите на иконку расширения на панели инструментов браузера
2. В открывшемся окне будет отображена информация о текущей вкладке и доступных пакетах модификаций
3. Для применения пакета модификаций к текущему сайту, включите соответствующий пакет
4. Для установки новых пакетов нажмите на кнопку "Магазин" и выберите интересующий вас пакет
5. Для создания собственного пакета модификаций нажмите на кнопку "Создать"
6. Для импорта пакета модификаций из файла или строкового кода нажмите на кнопку "Импорт"

Редактор пакетов модификаций

1. В редакторе пакетов вы можете создавать и редактировать пакеты модификаций
2. Укажите название, описание, иконку и сайты, на которых будет применяться пакет
3. Напишите CSS-код для изменения внешнего вида сайта
4. Напишите JavaScript-код для изменения функциональности сайта
5. Нажмите кнопку "Сохранить" для сохранения пакета

Маркетплейс пакетов

1. В маркетплейсе вы можете просматривать, устанавливать и оценивать пакеты, созданные другими пользователями
2. Для поиска пакетов используйте строку поиска
3. Для сортировки пакетов используйте кнопки сортировки
4. Для установки пакета нажмите кнопку "Установить" на карточке пакета
5. Для оценки пакета выберите количество звезд в его карточке
6. Для загрузки собственного пакета в маркетплейс нажмите кнопку "Загрузить" и выберите пакет из списка

Поддержка и обратная связь

Если у вас возникли вопросы или предложения по улучшению расширения, вы можете обратиться к разработчикам через GitHub-репозиторий проекта или по электронной почте.

ПРОМЕЖУТОЧНЫЙ ИТОГ

В представленном проекте была поставлена цель - разработка браузерного расширения EasyAccess, которое будет предоставлять возможности для повышения доступности веб-сайтов для людей с ограниченными возможностями здоровья, а также обеспечивать удобство и надежность для пользователей.

Цель проекта была достигнута благодаря реализации ряда задач:

- Проведено исследование предметной области, что помогло выявить основные потребности пользователей с ограниченными возможностями и особенности обеспечения доступности веб-ресурсов;
- Создана и реализована логическая схема базы данных с использованием PostgreSQL и TimescaleDB, обеспечивающая эффективное хранение и управление информацией о пакетах модификаций, пользователях и других сущностях;
- Разработан пользовательский интерфейс расширения с использованием TypeScript, React и Solid.js, обеспечивающий удобство использования и навигации;
- Реализована система авторизации и регистрации через Supabase, обеспечивающая безопасность персональных данных пользователей;
- Разработан функционал управления пакетами модификаций, включающий в себя создание, редактирование, импорт и экспорт пакетов;
- Создан маркетплейс пакетов модификаций, позволяющий пользователям обмениваться созданными пакетами;
- Реализован механизм инъекции CSS и JavaScript кода в веб-страницы для модификации их внешнего вида и функциональности;
- Обеспечена безопасность данных пользователей через механизмы аутентификации и авторизации, а также безопасного хранения данных;
- Проведено тестирование функционала и интерфейса, что позволило выявить и исправить ошибки.

Разработанное браузерное расширение EasyAccess представляет собой готовый к использованию продукт, который обеспечивает пользователей удобством и эффективностью. Однако для его дальнейшего развития и

совершенствования можно предложить ряд дополнительных функций, которые значительно улучшат опыт пользователей:

- Расширение поддержки для других браузеров, таких как Safari и Opera;
- Добавление функционала для работы с аудио-контентом, включая аудиодескрипцию и преобразование текста в речь;
- Интеграция с системами машинного обучения для автоматического анализа и улучшения доступности веб-страниц;
- Разработка API для интеграции с другими приложениями и сервисами;
- Реализация совместной работы над пакетами модификаций для командной разработки.

В процессе разработки браузерного расширения были освоены и применены современные технологии, такие как TypeScript, React, Solid.js, PostgreSQL, TimescaleDB и Supabase, что позволило значительно повысить производительность и надежность системы.

Таким образом, все поставленные в проекте задачи были успешно решены и цель проекта достигнута. Разработанное браузерное расширение EasyAccess предоставляет эффективный инструмент для повышения доступности веб-сайтов, что особенно важно для людей с ограниченными возможностями здоровья.