|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

|  |  |
| --- | --- |
| **ФАКУЛЬТЕТ** | ***ИУК «Информатика и управление»*** |
| **КАФЕДРА** | ***ИУК4 «Программное обеспечение ЭВМ,*** |
|  | ***информационные технологии»*** |

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовому проекту на тему:**

|  |
| --- |
| ***Разработка веб-приложения для сокращения ссылок*** |
|  |

|  |  |
| --- | --- |
| по дисциплине | *Кроссплатформенная разработка программного обеспечения* |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Студент гр. | *ИУК4-51Б* | |  |  | ( | *Отрошенко Т.В.* | ) |
|  | |  |  | (подпись) |  | (Ф.И.О.) |  |
| Руководитель | |  |  |  | ( | *Глебов С.А.* | ) |
|  | |  |  | (подпись) |  | (Ф.И.О.) |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Оценка руководителя |  | баллов |  |  |
|  | 30-50 |  | (дата) |  |
| Оценка защиты |  | баллов |  |  |
|  | 30-50 |  | (дата) |  |
| Оценка проекта |  | баллов |  | |
|  |  |  | (оценка по пятибалльной шкале) | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Комиссия: |  |  | ( | *Глебов С.А.* | ) |
|  |  | (подпись) |  | (Ф.И.О.) |  |
|  |  |  | ( | *Гришунов С.С.* | ) |
|  |  | (подпись) |  | (Ф.И.О.) |  |
|  |  |  | ( | *Гагарин Ю.Е.* | ) |
|  |  | (подпись) |  | (Ф.И.О.) |  |

Калуга, 2023

Калужский филиал   
федерального государственного бюджетного образовательного учреждения высшего образования

***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»   
(КФ МГТУ им. Н.Э. Баумана)***

УТВЕРЖДАЮ

Заведующий кафедрой *ИУК4*

( *Гагарин Ю.Е.)*

« *09* » *сентября* 20*23* г.

**ЗАДАНИЕ**

**на выполнение курсового проекта**

|  |  |
| --- | --- |
| по дисциплине | ***Кроссплатформенная разработка программного обеспечения*** |

|  |  |  |
| --- | --- | --- |
| Студент | *Отрошенко Т.В. ИУК4-51Б* | |
|  | | (фамилия, инициалы, индекс группы) |
| Руководитель | | *Глебов С.А.* |
|  | | (фамилия, инициалы) |

График выполнения проекта: 25% к *4* нед., 50% к *7* нед., 75% к *10* нед., 100% к *14*  нед.

|  |
| --- |
| ***1. Тема курсового проекта*** |
| Разработка веб-приложения для сокращения ссылок |
|  |
|  |
| ***2. Техническое задание*** |
| Разработать веб-приложение, позволяющее пользователю воспроизводить музыкальные |
| треки, создавать коллекции (плейлисты), а также получать рекомендации к выбранным |
| композициям |

|  |
| --- |
| ***3. Оформление курсового проекта*** |
| 3.1. Расчетно-пояснительная записка на  *48*  листах формата А4. |
| 3.2. Перечень графического материала КП (плакаты, схемы, чертежи и т.п.) |
| *1. Алгоритм работы приложения* |
| *2. Визуализация компонентов сайта* |
|  |
|  |

Дата выдачи задания « *09* » *сентября* *2023* г.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Руководитель курсового проекта | | |  | | / | *Глебов С.А.* |
|  | | (подпись) | | |  | (Ф.И.О.) |
| Задание получил |  | | / | *Отрошенко Т.В.* | / | « *09* » *сентября* 20*23* г. |
|  | (подпись) | |  | (Ф.И.О.) |  |  |

СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 3](#_Toc152499449)

[ВВЕДЕНИЕ 4](#_Toc152499450)

[1. МЕТОДЫ И ИНСТРУМЕНТЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ 5](#_Toc152499451)

[1.1. Техническое задание 5](#_Toc152499452)

[1.2. Анализ существующих аналогов 10](#_Toc152499453)

[1.3. Перечень задач, подлежащих решению в процессе разработки. 11](#_Toc152499454)

[1.4. Обоснование выбора СУБД 12](#_Toc152499455)

[1.5. Обоснование выбора языков программирования 14](#_Toc152499456)

[1.6. Обоснование выбора сред разработки 15](#_Toc152499457)

[1.7. Описание реализуемой архитектуры клиент-сервер 17](#_Toc152499458)

[1.8. Выбор средства для взаимодействия с базой данных 19](#_Toc152499459)

[Выводы 20](#_Toc152499460)

[2. ПРОЕКТИРОВАНИЕ КОМПОНЕНТОВ ПРОГРАММНОГО ПРОДУКТА 22](#_Toc152499461)

[2.1. Разработка структуры системы 22](#_Toc152499462)

[2.2. Разработка базы данных 22](#_Toc152499463)

[2.3. Структура Backend-приложения 27](#_Toc152499464)

[2.4. Разработка Frontend части 28](#_Toc152499465)

[2.5. Взаимодействие Frontend и Backend. 29](#_Toc152499466)

[2.6. Разработка рекомендательной системы 31](#_Toc152499467)

[2.7. Deploy проекта 34](#_Toc152499468)

[Выводы 34](#_Toc152499469)

[3. ТЕСТИРОВАНИЕ И ИНТЕГРАЦИЯ КОМПОНЕНТОВ ПРОГРАММНОГО ПРОДУКТА…. 35](#_Toc152499470)

[3.1. Тестирование системы 35](#_Toc152499471)

[3.2. Руководство разработчика Fronted (клиентской части) 36](#_Toc152499472)

[3.3. Руководство администратора 38](#_Toc152499473)

[3.4. Руководство пользователя 39](#_Toc152499474)

[Выводы 44](#_Toc152499475)

[ЗАКЛЮЧЕНИЕ 45](#_Toc152499476)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 46](#_Toc152499477)

[Основная литература 46](#_Toc152499478)

[Дополнительная литература 46](#_Toc152499479)

ВВЕДЕНИЕ

Сокращение ссылок является важным инструментом в современном интернете. Это процесс заменены длинных URL-адресов на более короткие и удобные ссылки. Если речь идет о значимости сокращения ссылок, есть несколько важных аспектов, которые стоит упомянуть.

1. Удобство использования: Длинные URL-адреса могут быть громоздкими и сложными для запоминания и ввода. Сокращение ссылок позволяет создавать более короткие и запоминающиеся URL-адреса, которые легко передавать и использовать.

2. Экономия места: Сокращение ссылок особенно полезно в контексте социальных сетей и сообщений с ограничением символов, таких как Twitter. Более короткие ссылки занимают меньше места, освобождая пространство для дополнительного текста.

3. Улучшение внешнего вида: Сокращение ссылок позволяет создавать более эстетичные и профессионально выглядящие URL-адреса. Более чистые и удобочитаемые ссылки могут повысить восприятие и доверие пользователей.

4. Отслеживание и аналитика: Сокращение ссылок позволяет отслеживать и анализировать клики и действия пользователей. Это позволяет получить данные о количестве кликов, источниках трафика и других метриках, что важно для оценки эффективности маркетинговых кампаний и веб-страниц.

5. Скрытие длинных и сложных URL-адресов: Длинные URL-адреса могут выглядеть неэстетично и вызывать недоверие у некоторых пользователей. Сокращение ссылок позволяет скрыть сложные URL-адреса за короткими и более понятными ссылками, что может повысить кликабельность и доверие к ссылкам.

6. Перенаправление и обновление ссылок: Сокращенные ссылки позволяют легко перенаправлять пользователей с одного адреса на другой. В случае изменения URL-адресов, использование сокращенных ссылок упрощает обновление ссылок на веб-страницах, в электронных письмах и в других местах.

Сокращение ссылок имеет большое значение в современном интернете. И даже в будущем их актуальность будет только расти, так как объем и значимость информации в интернете будет увеличиваться.

Объектом работы выступает максимально простой и доступный сервис по сокращению ссылок.

Целью проекта выступает создание веб-приложения, предоставляющего возможность изменения ссылок.

Задачи можно описать следующим образом:

1. Исследование существующих методов сокращения ссылок.

2. Проектирование архитектуры. Необходимо определить компоненты системы, их взаимодействие и функциональные возможности. Выбор способа хранения ссылок, определение способов генерации коротких ссылок.

3. Разработка пользовательского интерфейса.

4. Реализация алгоритма сокращения ссылок.

5. Реализация функционала переходов по сокращенным ссылкам.

7. Тестирование и отладка системы.

8. Оформление работы и написание отчета.

1. МЕТОДЫ И ИНСТРУМЕНТЫ ПРОГРАММНОЙ ИНЖЕНЕРИИ

1.1. Техническое задание

Наименование системы

Настоящее Техническое задание определяет требования и порядок создания «Веб-приложения для сокращения ссылок».

Основания для разработки

С развитием интернета и увеличением количества веб-сайтов, стало сложно запоминать и передавать длинные URL-адреса. Сокращение ссылок позволяет заменять длинные URL-адреса более короткими и более удобными для использования. Короткие ссылки легко передаются и запоминаются, что делает их удобными для обмена в социальных сетях, мессенджерах и других онлайн-платформах. Также сокращение ссылок может помочь улучшить внешний вид сообщений и уменьшить количество символов, необходимых для набора.

Исполнители

Исполнителем проекта является студент Калужского филиала МГТУ им. Н. Э. Баумана, факультета ИУК, группы ИУК4-51Б, Отрошенко Таисия Владимировна.

Краткая характеристика области применения

Веб-приложение будет применяться пользователями сети интернет в основном в областях, связанных с созданием описательного текста. Таких как социальные медиа, рекламные кампании, академические и исследовательские работы.

Целевая аудитория

Авторы и редакторы статей в сети интернет.

Назначение приложения

Сокращенные ссылки призваны сделать процесс обмена, запоминания и отслеживания URL-адресов более эффективным и надежным.

Цели создания

Основная цель разработки приложения по сокращению ссылок заключается в улучшении пользовательского опыта.

Плановые сроки начала и окончания работы по созданию системы

Планируемые сроки начала и окончания работы над проектом: 09.09.2023 – 17.12.2023.

Требования к веб-приложению

Функциональные требования:

1. Приложение должно предоставлять возможность пользователю ввода оригинальной ссылки.
2. Приложение должно генерировать уникальный короткий код для каждой сокращенной ссылки.
3. Приложение должно сохранять сокращенную ссылку и ее оригинальный URL для дальнейшего использования.
4. Приложение должно предоставлять возможность пользователю просмотреть список всех созданных им сокращенных ссылок.
5. Пользователь должен иметь возможность удалить сокращенную ссылку.

Нефункциональные требования:

* Приложение должно быть легким и быстрым в использовании.
* Приложение должно быть достаточно надежным и обеспечивать безопасность пользовательских данных, таких как оригинальные URL и сокращенные ссылки.
* Приложение должно быть доступно для работы на разных платформах, включая веб-браузеры и мобильные устройства.
* Приложение должно быть масштабируемым, чтобы удовлетворять возрастающим требованиям пользователей.

Требования к надежности

1. Устойчивость функционирования

* Гарантированная доступность приложения
* Обеспечение бесперебойного функционирования

2. Организация бесперебойного питания технических средств

* Использование собственного сервера с гарантированным временем работы

3. Регулярное выполнение рекомендаций и норм времени

* Соблюдение рекомендаций и норм времени, предложенных Минтрудом РФ и другими организациями для обеспечения эффективного технического обслуживания и сопровождения программного обеспечения.

Требования к программной документации

Должны быть разработаны следующие программные документы:

1. Расчетно-пояснительная записка:

* Техническое задание;
* Научно-исследовательская часть;
* Проектная часть;
* Производственно-технологическая часть;

1. Графическая часть - 2 листа формата А3 включающие в себя:

* структурную схему;
* демонстрационный чертеж;

Стадии разработки

Техническое задание

1. Обоснование перспективности реализуемого проекта:

* постановка задачи;
* сбор базовых материалов;
* установка критериев системы;
* необходимость проведения исследовательских работ;

1. Исследовательская работа:

* выбор оптимальных методов решения поставленной задачи;
* определение требований к техническим средствам;
* обоснование практической возможности реализации данного проекта;

1. Разработка и утверждение технического задания:

* определение требований к проекту;
* определение стадий, этапов и сроков разработки проекта и документации на нее;
* согласование и утверждение технического задания;

Технический проект

1. Разработка технического проекта:

* определение формы представления входных и выходных данных;
* определение конфигурации технических средств;

1. Утверждение технического проекта:

* установка плана по разработке проекта;
* создание пояснительной записки;
* утверждение технического проекта;

Этапы разработки

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии проектирования программы должен быть выполнен этап выборки программного обеспечения, библиотек для создания, этап проектирования системы в целом, разработка рабочей документации.

На стадии реализации производится разработка и тестирование спроектированной программы.

**1.2.** **Анализ существующих аналогов**

В настоящее время существуют аналоги, предоставляющие подобный функционал. Однако данные системы являются перегруженными, так как в дополнение к актуальному перечню задач предоставляют множество других, которые зачастую являются избыточными. Самые используемые существующие приложения: «Я Кликер», не позволяющий хранить и анализировать использование ссылки, и «GOO.su» с избыточным функционалом.

1.3. Перечень задач, подлежащих решению в процессе разработки.

1. Определение оптимальной архитектуры системы:

* Выбор архитектуры, учитывающей потребности взаимодействия между сервером и клиентами.
* Разработка структуры системы, включая серверную часть и клиентскую часть в виде веб-приложения.

2. Средства реализации программного продукта:

* Выбор и настройка сред разработки и операционной системы для эффективного процесса разработки и развертывания.

3. Реализация серверной части:

* Написание серверного приложения для обработки запросов от клиентов.
* Разработка API для взаимодействия с хранилищем данных.

4. Реализация клиентской части:

* Создание веб-приложения для удобного взаимодействия пользователей с сервисом.

5. Взаимодействие с хранилищем данных:

* Оптимизация структуры данных для эффективного хранения и быстрого доступа к информации.
* Реализация механизмов взаимодействия сервера с хранилищем данных для обеспечения оперативности ответов на запросы клиентов.

6. Разработка документации:

* Создание технической документации по архитектуре, интерфейсам и инструкциям по развертыванию.
* Подготовка руководства пользователя для использования сервиса.

1.4. Обоснование выбора СУБД

JSON файл: JSON (JavaScript Object Notation) — это формат хранения и передачи данных, основанный на JS-объектах. Он очень популярен в веб-разработке, так как удобен для чтения и записи данных. Использование JSON файла для хранения данных ссылок позволяет легко расширять функционал приложения добавлением, обновлением и удалением ссылок без необходимости использования базы данных.

1.5. Обоснование выбора языков программирования

JavaScript (JS): JS является одним из самых популярных языков программирования в веб-разработке. Он широко используется для создания клиентских и серверных приложений. В данном случае, JS будет использован для написания логики работы приложения по сокращению ссылок на клиентской стороне.

React: React — это JavaScript-библиотека для разработки пользовательских интерфейсов. Она позволяет эффективно организовать работу с отображением данных, обновление DOM и взаимодействие пользователей с веб-приложением. В данном проекте можно использовать React для создания интерфейса пользователя, отображения текущих ссылок и управления ими.

Express.js — это популярный фреймворк для разработки серверной части на языке JavaScript. Она предоставляет набор функций и middleware для обработки маршрутов, запросов и других задач разработки сервера. Express.js обладает удобным и интуитивным интерфейсом, позволяя быстро создавать серверные приложения и обрабатывать запросы от клиентов. Благодаря простоте и гибкости, Express.js позволяет разработчикам создавать эффективный и масштабируемый серверный код.

1.6. Обоснование выбора сред разработки

VS Code: это популярный и мощный редактор кода, разработанный Microsoft. Он предоставляет широкий набор функций и инструментов, которые делают процесс разработки более эффективным и удобным. VS Code поддерживает большое количество расширений, позволяющих настроить редактор под конкретные потребности разработчика. Благодаря его простому интерфейсу, быстрому отклику и возможности работать с различными языками программирования, VS Code является одним из самых популярных выборов среди разработчиков. Он подходит для разработки как веб-приложений, так и других типов проектов.

1.7. Описание реализуемой архитектуры клиент-сервер

Разрабатываемое приложение предполагает взаимодействие с данными, согласно архитектуре клиент-сервер.

Архитектура клиент-сервер представляет собой распределенную структуру, основными элементами которой являются: сервер, выполняющий запросы и совокупность клиентов, инициализирующих эти самые запросы. Связь между сервером и клиентами при этом организуется с использованием определенного протокола. Сервер хранит ресурсы, необходимые клиентам.

В рамках разрабатываемой системы, сервер хранит базу данных с информацией о ссылках. Сервер принимает, обрабатывает запросы от клиентов и отправляет ответ. Клиенты – пользователи приложения, формирующие запросы к серверу на подачу, обработку и сохранение данных, а также отображения информации на экране.

Выводы

Таким образом исходя из требований к реализуемой системе, рассмотрения возможностей наиболее подходящих инструментов было решено использовать следующие компоненты:

* Для Backend (серверной части) были выбраны следующие инструменты: JavaScript, Express.js.
* Для Frontend (визуальной части) была выбрана библиотека React.js
* Базируясь на требованиях к системе, было принято решение использовать JSON, как самый доступный формат хранения и удобный в соединении с JavaScript. Эта интеграция обеспечивает эффективную передачу и обработку данных между серверными и клиентскими компонентами приложений, что способствует ускорению разработки и облегчению взаимодействия между различными модулями программного обеспечения, основанными на JavaScript.

2. ПРОЕКТИРОВАНИЕ КОМПОНЕНТОВ ПРОГРАММНОГО ПРОДУКТА

2.1. Разработка структуры системы

Разрабатываемая система состоит из следующих компонентов:

Файл данных – хранит всю необходимую информацию о ссылках.

Серверная часть – представляет собой компьютер-сервер с установленной OC Windows, развернутом Express.js приложении.

Клиентская часть – веб-приложение, доступное в любом браузере по URL, позволяющее пользователям получать доступ к интерфейсу взаимодействия с сервером.

2.2. Разработка базы данных

Данные о ссылках хранятся в виде следующей структуры:

|  |  |
| --- | --- |
| Наименование поля | Тип данных |
| Оригинальная ссылка | String |
| Уникальный код | String |
| Счетчик переходов по ссылке | Number |
| Дата создания | Date |

2.3. Структура клиент-серверного взаимодействия

Структура клиент-серверного взаимодействия с использованием HTTP протокола реализована следующим образом:

1. Клиент отправляет POST-запрос по адресу '/generate' с параметром 'originalUrl' в теле запроса. Сервер принимает запрос и извлекает значение 'originalUrl' из тела запроса. Проверяет, является ли полученный 'originalUrl' валидным URL при помощи функции 'validUrl.isUri()'. Если 'originalUrl' является валидным URL, то генерируется уникальный идентификатор 'urlId' при помощи функции 'shortId.generate()'. Сервер отправляет ответ клиенту с 'urlId'.

2. Клиент отправляет POST-запрос по адресу '/insert' с параметрами 'originalUrl' и 'shortUrl' в теле запроса. Сервер принимает запрос и извлекает значения 'originalUrl' и 'shortUrl' из тела запроса. Создается объект 'urlObject' с полями 'originalUrl', 'shortUrl', 'clicks' (количество переходов по ссылке, инициализируется нулем) и 'date' (текущая дата). Считывается содержимое файла базы данных 'dbData'. Добавляется объект 'urlObject' в массив 'urls' в 'dbData'. Записывается обновленное содержимое 'dbData' в файл базы данных .

3. Клиент отправляет GET-запрос по адресу '/read'. Сервер считывает содержимое файла базы данных 'dbData'. Отправляет клиенту массив 'urls' из объекта 'dbData'.

4. Клиент отправляет GET-запрос по адресу '/:route' с параметром 'route' - сокращенный URL. Сервер считывает содержимое файла базы данных 'dbData'. Ищет объект 'url' в массиве 'urls' в 'dbData', у которого значение поля 'shortUrl' равно полученному параметру 'route'. Если найден объект 'url', то увеличивает значение поля 'clicks' на единицу и записывает обновленное 'dbData' в файл базы данных 'dbFilePath'. Перенаправляет клиента на оригинальный URL, указанный в поле 'originalUrl' объекта 'url'. Если объект 'url' не найден, сервер отправляет клиенту статус 404 и сообщение 'No url found'.

5. Клиент отправляет POST-запрос по адресу '/delete/:id' с параметром 'id' - сокращенный URL, который нужно удалить. Сервер извлекает значение параметра 'id' из запроса. Считывается содержимое файла базы данных 'dbData'. Находится индекс объекта 'url' в массиве 'urls' в 'dbData', у которого значение поля 'shortUrl' равно полученному параметру 'id'. Если индекс не равен -1 (т.е. объект найден), то удаляется объект 'url' из массива 'urls' в 'dbData'. Обновленное содержимое 'dbData' записывается в файл базы данных 'dbFilePath'. Сервер отправляет клиенту ответ 'URL deleted successfully' или статус 404 и сообщение 'URL not found'.

Ниже приведена блох-схема, иллюстрирующая алгоритм получения информации от сервера.

Выводы

В результате были выполнены проектирование и реализация компонентов программного продукта: базы данных, серверной части и клиентской части. Функциональные требования технического задания были учтены.

3. ТЕСТИРОВАНИЕ И ИНТЕГРАЦИЯ КОМПОНЕНТОВ ПРОГРАММНОГО ПРОДУКТА

3.1. Тестирование системы

Тестирование системы производилось с использованием персонального компьютера. Клиентская и серверная части приложения запускались с одного устройства. Для тестирования системы необходимо запустить сервер, клиента и перейти по URL сервера.

Требования к веб-приложению

Для работы веб-приложения необходимо, чтобы компьютер обладал следующими характеристиками:

Для использования приложения, компьютеру необходимо соответствовать определенным техническим требованиям. Основные из них:

1. Процессор: Компьютер должен иметь процессор с поддержкой JavaScript и достаточной производительностью для обработки кода в реальном времени.

2. Операционная система: Приложение может быть запущено на разных операционных системах, таких как Windows, macOS, Linux и т. д. Компьютер должен работать на совместимой операционной системе.

3. Браузер: Приложения выполняются в среде браузера. Компьютер должен иметь установленный современный и совместимый веб-браузер, такой как Google Chrome, Mozilla Firefox, Safari или Microsoft Edge.

4. Память и хранение: Компьютеру требуется минимальное количество оперативной памяти (RAM), достаточное для запуска и функционирования приложения.

5. Интернет-соединение: Требуется подключение к интернету для загрузки ресурсов, выполнения запросов к серверам и обновления данных. Компьютер должен иметь стабильное и достаточно быстрое интернет-соединение.

3.4. Руководство пользователя

После перехода по URL сервера клиенту сразу доступен весь функционал приложения.

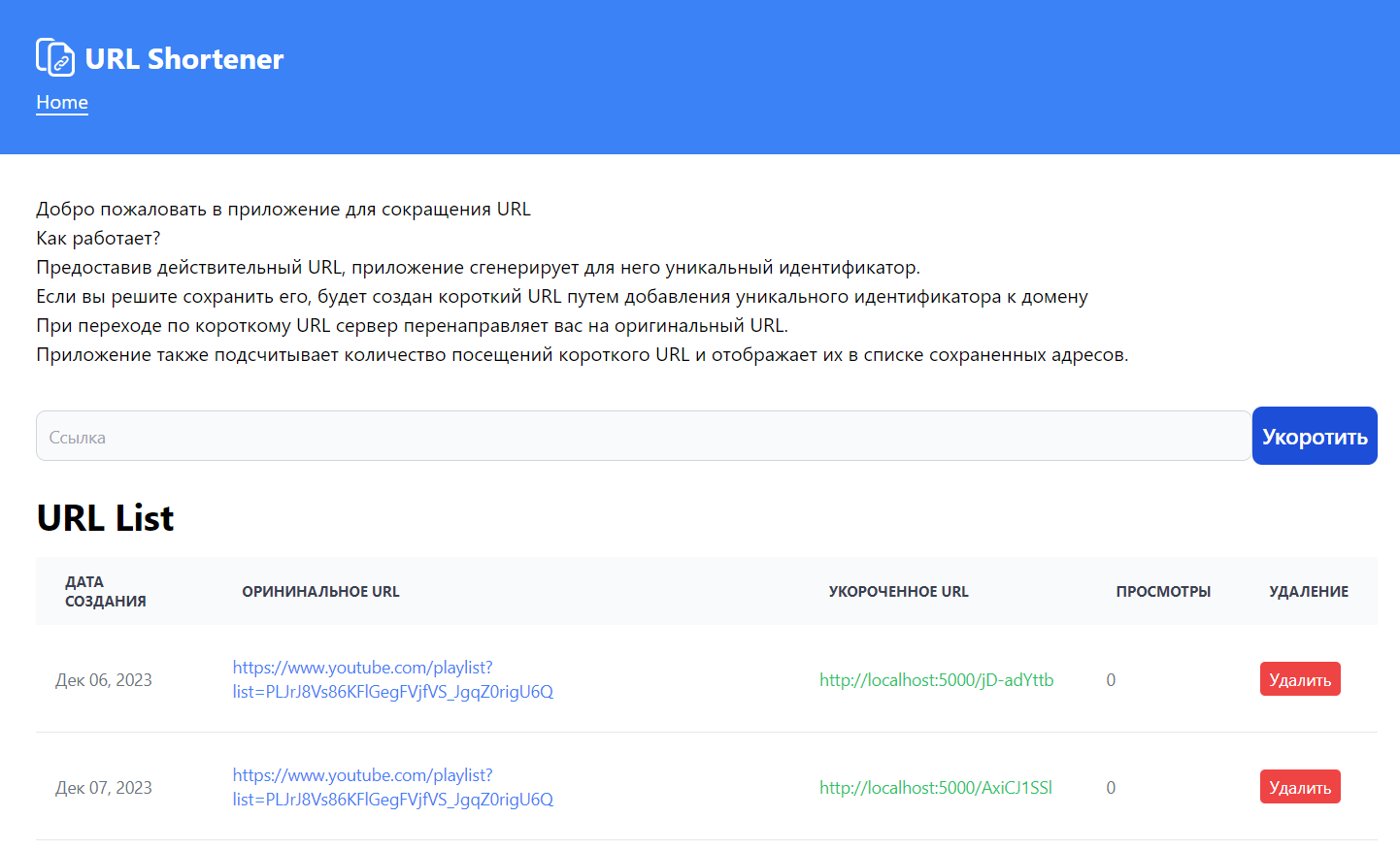


Рис. 1. Общий вид приложения

Пользователя встречает вводная информация о приложении, после которой располагается поле для ввода ссылки

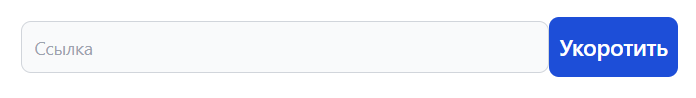


Рис. 2. Поле ввода данных пользователя

При некорректных введенных данных выводится ошибка.

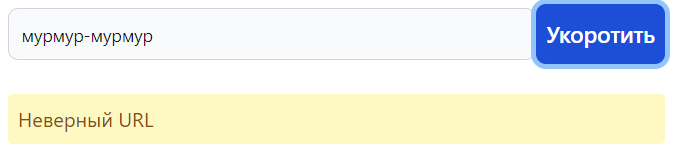


Рис. 3. Информационное сообщение при некорректных данных

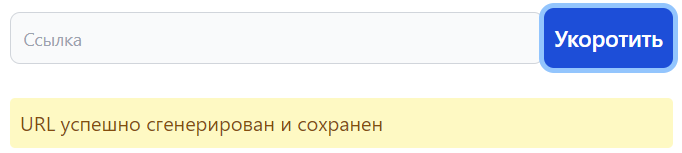


Рис. 4. Информационное сообщение при корректных данных

Поле ввода остается открытым для введения новых данных. После нажатия кнопки «Укоротить» при успешной генерации в нижнем блоке страницы, где располагаются данные, появляется информация об обработанной ссылке. По порядку: дата создания, оригинальный URL, короткий URL, счетчик переходов и кнопка удаления.

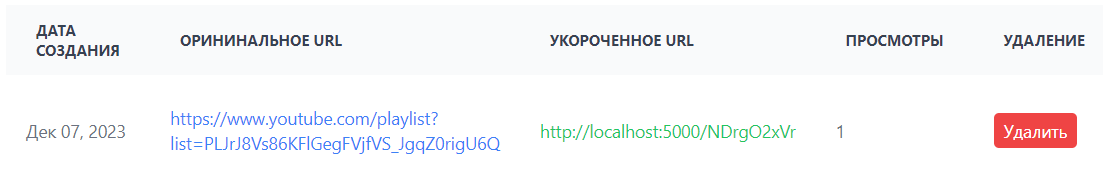


Рис. 5. Блок просмотра данных сервера

При нажатии кнопки «Удалить» запись пропадает с экрана и удаляется из базы данных.

Выводы

В результате были выполнены тестирование компонентов программного продукта. Разработана техническая документация.

ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы был реализован стриминговый веб-сервис для прослушивания музыки.

Реализация музыкального веб-сервиса, основанного на технологиях React, Flask, SQLAlchemy и PostgreSQL, представляет собой современное и эффективное решение для создания платформы, предназначенной для прослушивания и управления музыкальным контентом. В ходе разработки проекта были определены ключевые этапы и решены ряд задач с учетом требований к функциональности и надежности системы.

Музыкальный веб-сервис, представленный в данном проекте, сочетает в себе современные технологии, удобство использования и высокий уровень функциональности. Он ориентирован на предоставление уникального и приятного опыта прослушивания музыки, а его архитектурные решения делают его гибким и готовым к дальнейшему развитию и улучшению.

Проект подлежит дальнейшей доработке, внедрению новых функций, реализации новых программных идей.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Основная литература

1. Романенко, В.В. Объектно-ориентированное программирование : учебное пособие / В.В. Романенко ; Министерство образования и науки Российской Федерации, Томский Государственный Университет Систем Управления и Радиоэлектроники (ТУСУР). - Томск : Томский государственный университет систем управления и радиоэлектроники, 2014. - 475 с. : ил. - Библиогр.: с. 442. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=480517>.
2. Николаев, Е.И. Объектно-ориентированное программирование : учебное пособие / Е.И. Николаев ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет». - Ставрополь : СКФУ, 2015. - 225 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=458133>.
3. Суханов, М.В. Основы Microsoft .NET Framework и языка программирования C# : учебное пособие / М.В. Суханов, И.В. Бачурин, И.С. Майоров ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования Северный (Арктический) федеральный университет им. М.В. Ломоносова. - Архангельск : ИД САФУ, 2014. - 97 с. : схем., табл., ил. - Библиогр. в кн. - ISBN 978-5-261-00934-4 ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=312313>.

Дополнительная литература

1. Мейер, Б. Объектно-ориентированное программирование и программная инженерия [Электронный ресурс] / Б. Мейер. — 3-е изд. — Электрон. текстовые данные. — М. : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Эр Медиа, 2019. — 285 c. — 978-5-4486-0513-0. — Режим доступа: <http://www.iprbookshop.ru/79706.html>