



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 курсовой работе на тему:

Разработка приложения досок Scrum-команды

по дисциплине Базы данных

Студент гр. ИУК4-62Б _____ (подпись) (Карельский М.К.)
(Ф.И.О.)

Руководитель _____ (подпись) (Глебов С.А.)
(Ф.И.О.)

Оценка руководителя _____ баллов _____
30-50 (дата)

Оценка защиты _____ баллов _____
30-50 (дата)

Оценка работы _____ баллов _____
(оценка по пятибалльной шкале)

Комиссия: _____ (подпись) (Глебов С.А.)
(Ф.И.О.)

_____ (подпись) (Гришунов С.С.)
(Ф.И.О.)

_____ (подпись) (Гагарин Ю.Е.)
(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 _____

_____ (_____)

«____» _____ 20__ г.

**ЗАДАНИЕ
на выполнение курсовой работы**

по дисциплине Базы данных

Студент Карельский М.К. ИУК4-62Б

(фамилия, инициалы, индекс группы)

Руководитель Глебов С.А.

(фамилия, инициалы)

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

1. Тема курсовой работы

Разработка приложения досок Scrum-команды

2. Техническое задание

Разработать базу данных, ранние версии клиентской и серверной части приложения
досок Scrum-команды

3. Оформление курсовой работы

3.1. Расчетно-пояснительная записка на 36 листах формата А4.

3.2. Перечень графического материала КР (плакаты, схемы, чертежи и т.п.)

1. Структура БД

2. Демонстрационный чертеж

Дата выдачи задания «09» февраля 2023 г.

Руководитель курсовой работы _____

(подпись)

/

Глебов С.А

(Ф.И.О.)

Задание получил _____

(подпись)

/

Карельский М.К.

(Ф.И.О.)

/

«09» февраля 2023 г.

РЕФЕРАТ

Расчетно-пояснительная записка 43 с., 27 рисунков, 3 таблицы, 19 источников.

Разработка веб-сервиса обработки мультимедиа.

Объектом разработки является система, позволяющая проводить простую обработку изображений, аудио и видео без необходимости установки программного обеспечения со стороны пользователя.

Цель проекта – разработка веб-сервиса обработки изображений, видео и аудио с помощью реализации брокера сообщений.

Поставленные задачи решаются путем разработки серверной и клиентских частей, а также организации их взаимодействия.

СОДЕРЖАНИЕ

РЕФЕРАТ	3
СОДЕРЖАНИЕ	4
ВВЕДЕНИЕ	5
1. АНАЛИЗ ТРЕБОВАНИЙ И ТЕХНОЛОГИЙ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	6
1.1. Основные требования к разрабатываемой системе	6
1.2. Анализ аналогов и прототипов	7
1.3. Обоснование выбора инструментов и платформы для разработки серверной части	10
1.4. Обоснование выбора инструментов и платформы для разработки клиентской части	14
2. ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	18
2.1. Разработка структуры системы	18
2.2. Структура базы данных	20
2.3. Описание API	23
3. КОНТРОЛЬ КАЧЕСТВА И ИНТЕГРАЦИЯ КОМПОНЕНТОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	26
3.1. Интерфейс клиентской части	26
3.2. Руководство пользователя	28
ЗАКЛЮЧЕНИЕ	40
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	41
Основная литература	41
Дополнительная литература	43

ВВЕДЕНИЕ

Актуальность темы курсового проекта обусловлена тем, что множество пользователей ежедневно сталкивается с необходимостью быстрого изменения своих файлов, например, изменение формата изображения или извлечение фрагмента из видео. Для решения подобных задач применяется различное настольное программное обеспечение. Однако описанные задачи могут возникать перед пользователем не так часто, чтобы у того появлялась необходимость в установке на свой ПК дополнительных программ, особенно если сама задача не является сложной, комплексной. В таком случае гораздо удобнее воспользоваться специальным онлайн-сервисом, предоставляющим возможность загрузки личных файлов мультимедиа, их последующей требуемой обработки и скачивания готового результата.

Объектом курсового проекта является разработка клиент-серверного веб-приложения.

Предметом исследования курсового проекта являются различные алгоритмы обработки мультимедиа.

Целью практики является разработка серверной части сервиса обработки изображений, видео и аудио с помощью реализации брокера сообщений, а также разработка клиентской части.

Для достижения поставленной цели решаются следующие задачи:

1. Реализовать алгоритмы обработки;
2. Создать базу данных;
3. Разработать API сервиса;
4. Интегрировать брокер сообщений;
5. Реализовать клиентскую часть;
6. Провести контейнеризацию.

1. АНАЛИЗ ТРЕБОВАНИЙ И ТЕХНОЛОГИЙ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

1.1. Основные требования к разрабатываемой системе

Система путем взаимодействия пользователя с веб-сайтом должна предоставлять ему следующие методы обработки мультимедиа:

1. Обработка изображений:

- Добавление водяных знаков;
- Сжатие;
- Конвертация в другой формат;
- Кадрирование;
- Отражение по горизонтали или вертикали;
- Изменение размеров;
- Поворот на заданный угол;

2. Обработка аудио:

- Извлечение фрагмента;
- Конвертация в другой формат;
- Добавление линейных переходов (затуханий);
- Изменение громкости каналов;
- Объединение двух аудио;
- Обратное воспроизведение;
- Перевод в одноканальный формат (моно);
- Изменение скорости воспроизведения.

3. Обработка видео:

- Перевод в черно-белое отображение;
- Извлечение фрагмента;
- Конвертация в другой формат;
- Извлечение аудиодорожки;

- Добавление линейных переходов (затуханий);
- Отражение по горизонтали или вертикали;
- Изменение громкости;
- Объединение двух видео;
- Обратное воспроизведение;
- Поворот на заданный угол;
- Изменение скорости воспроизведения.

1.2. Анализ аналогов и прототипов

Рассмотрим различные аналоги системы:

I LoveIMG.com

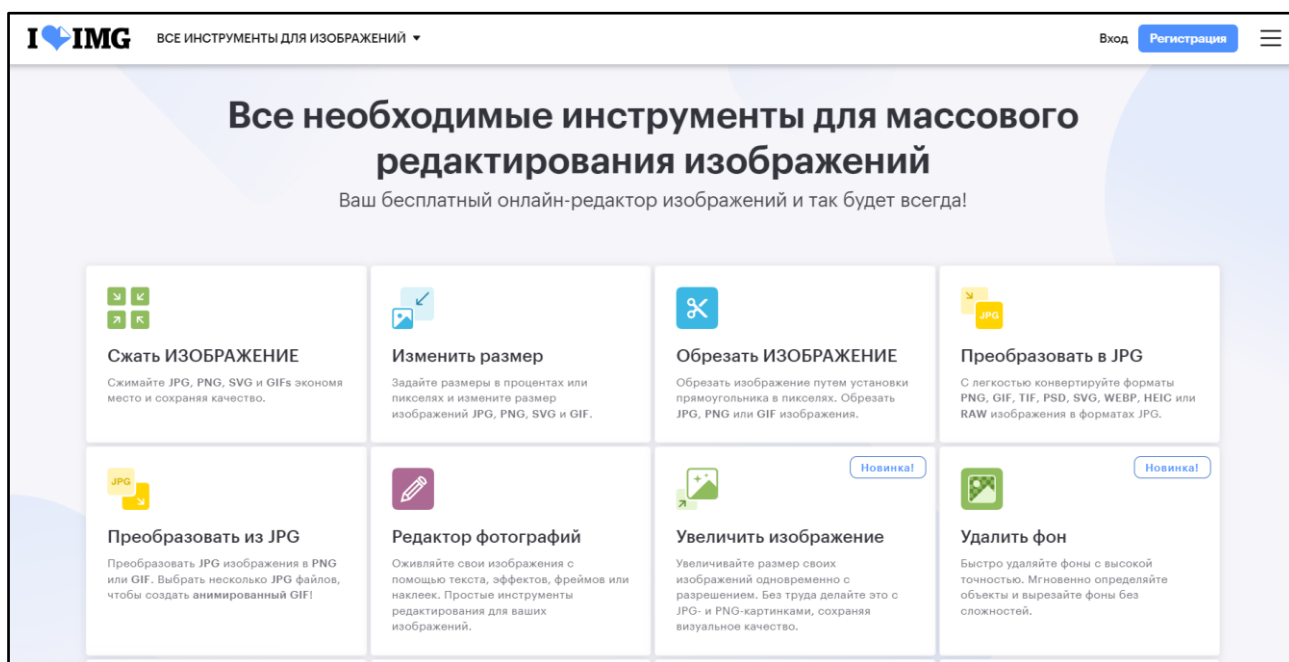


Рис. 1. Главная страница сайта I LoveIMG.com

I LoveIMG.com (см. рис. 1) предоставляет различные инструменты для редактирования изображений. В их число входит сжатие, кадрирование, поворот, добавление водяных знаков и преобразование в различные форматы.

Основной функционал доступен бесплатно. Имеется платная подписка стоимостью 234 руб./месяц, предоставляющая более широкие возможности и неограниченную обработку. Также предлагаются индивидуальные планы для бизнеса.

Convertio.co

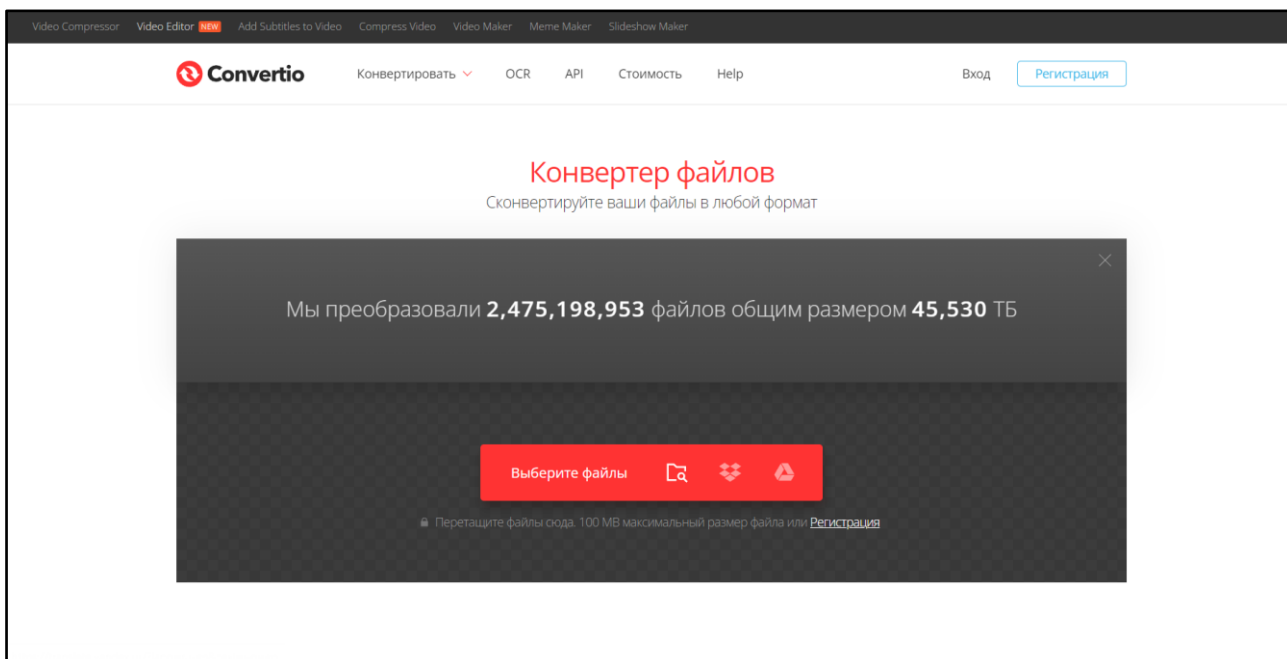


Рис. 2. Главная страница сайта Convertio.co

Convertio.co (см. рис. 2) – онлайн-сервис, предоставляющий возможность преобразовывать изображения, видео и аудио во множество различных форматов.

Бесплатная версия имеет ограничение максимального размера загружаемого файла – 100 МБ. Сервис предоставляет 3 уровня подписки: за 6\$, 9\$ и 16\$ в месяц. Более высокий уровень предоставляет возможность загрузки файлов большего размера, большего количества одновременных конвертаций, а также более высокий приоритет в очереди обработки.

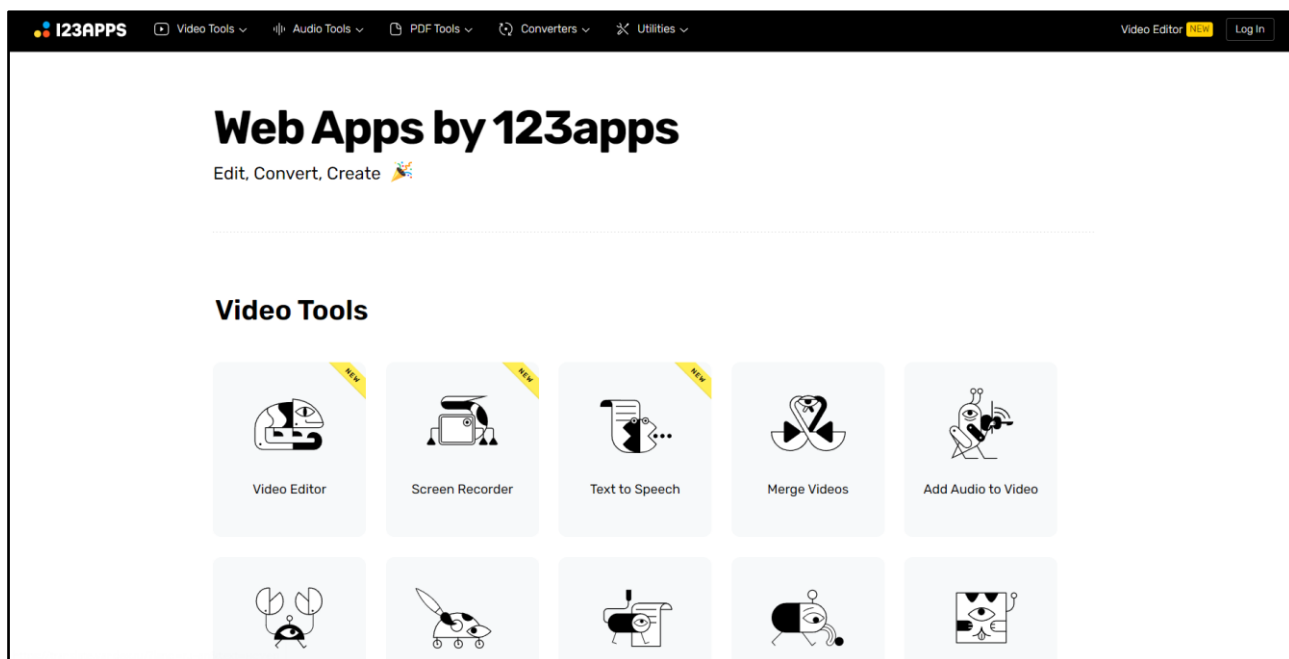


Рис. 3. Главная страница сайта 123apps.com

Сервис (см. рис. 3) предоставляет широкий спектр веб-приложений для обработки видео и аудио. Среди них: объединение, сокращение, кадрирование, поворот, отражение, изменение громкости и скорости видео, объединение, сокращение, изменение громкости и скорости аудио. Также доступны конвертеры изображений, видео и аудио.

Бесплатная версия накладывает ограничение на максимальный размер файла в 500 МБ и количество доступных обработок в день – 5. Предлагается подписка за 300 руб./месяц, снимающая ограничения на количество обработок в день и увеличивающая максимальный размер загружаемого файла до 4 ГБ.

1.3. Обоснование выбора инструментов и платформы для разработки серверной части

Язык программирования:

PHP

PHP используют около 80% всех сайтов, а самому языку уже более 25 лет. Также обладает широким сообществом, благодаря чему язык прост для обучения и получает постоянные обновления. Его достаточно легко установить и настроить.

Java

Java – один из самых популярных языков программирования. Используется уже более 20 лет и имеет широкую поддержку сообществом. Также обладает хорошей универсальностью, которая обеспечивается с помощью виртуальной машины Java (JVM).

C#

C# является широко известным и используемым объектно-ориентированным языком программирования. Для реализации серверной части используется достаточно популярный, стабильный и надежный фреймворк ASP.NET. Поддержку ASP.NET 6.0 обеспечивает среда разработки Visual Studio 2022.

Python

Python обладает большой популярностью среди различных сфер разработки, включая разработку серверов. Наиболее популярным фреймворком для этого является Flask, обладающий большим сообществом и позволяющий написать серверную часть достаточно просто. Помимо этого, Python может

применяться для написания самих обработчиков задач, так как имеет для этого множество полезных и удобных библиотек, например, Pillow.

Брокер сообщений:

RabbitMQ

RabbitMQ – это брокер распределенных сообщений, который собирает потоковые данные из нескольких источников и маршрутизирует их в разные пункты назначения для обработки. RabbitMQ обеспечивает легкость разработки, так как имеет библиотеки для множества языков, простое администрирование тонкую настройку.

Kafka

Представляет из себя распределённый программный брокер сообщений с открытым исходным кодом. Ядром функциональности является запись данных, хранение их в течение заданного времени и выдача этих данных по запросу. Предоставляет большую пропускную способность, возможность читать множество сообщений за раз, а также позволяет перечитывать ранее прочитанные сообщения.

SQS

SQS – сервис от Amazon, принимающий очереди сообщений для хранения. Весьма популярен за рубежом, предоставляет высокую безопасность и интеграцию со всеми сервисами компании. Однако в связи с полной зависимостью от Amazon возникают проблемы при появлении необходимости в переходе на другого поставщика.

База данных:

OracleDB

OracleDB – объектно-реляционная система управления базами данных компании Oracle. В связи с поддержкой крупной компании имеет высокую

надежность. Обеспечивает хорошую масштабируемость, безопасность и скорость. Однако также имеет высокую стоимость.

MySQL

MySQL – реляционная система управления базами данных с открытым исходным кодом. Имеет достаточно широкую популярность и сообщество. Предлагает простую установку и использование. Хорошо работает с данными на базовом уровне, однако в процессе масштабирования может потребоваться дополнительная поддержка, также имеющая высокую стоимость.

PostgreSQL

PostgreSQL – реляционная база данных с открытым кодом, является одной из наиболее известных среди всех существующих реляционных баз данных. Предоставляет хорошую масштабируемость, дополнительную защиту посредством использования ролей и прав. Полностью бесплатна, поддерживается сообществом, хорошо справляется с большими нагрузками.

MongoDB

MongoDB – документоориентированная система управления базами данных, не требующая описания схемы таблиц. Считается одним из классических примеров NoSQL-систем, использует JSON-подобные документы и схему базы данных. Написана на языке C++, что позволяет легко портировать ее на различные платформы.

Контейнеризатор:

Docker

Docker – это платформа для контейнеризации приложений с открытым исходным кодом. Она позволяет упаковывать приложения с их окружениями и зависимостями в контейнеры, а затем предоставляет возможность с помощью встроенных команд управлять ими. Предоставляет изолированное окружение

для каждого контейнера, каждый из которых можно легко переносить между различными средами. Обеспечивает эффективное использование ресурсов хостовой системы и возможность быстрого развертывания приложений.

Podman

Podman – это контейнеризатор с открытым исходным кодом, представляет собой утилиту командной строки. Позволяет запускать и управлять контейнерами без наличия демона. Обеспечивает изоляцию каждого контейнера с более высокой безопасностью.

Containerd

Containerd – бывшая часть Docker, реализует исполняемую среду для запуска контейнеров. Предоставляет минимальный набор функций для управления образами, а также для запуска и остановки контейнеров. Предлагает высокую производительность, стабильность и широкую поддержку сообщества.

Итоговый выбор:

В качестве языка программирования был выбран Python, используемый в написании как серверной части посредством фреймворка Flask, так и самих обработчиков задач. Для распределения сообщений был выбран RabbitMQ, предоставляющий легкую интеграцию в проект. Хранение данных производится в PostgreSQL, обладающий полным необходимым функционалом. Для контейнеризации используется наиболее распространенный Docker.

1.4. Обоснование выбора инструментов и платформы для разработки клиентской части

Фреймворк:

Angular

Популярный и весьма востребованный фреймворк, имеющий официальный набор функциональных компонентов для создания пользовательских интерфейсов. Направлен на создание одностраничных приложений, что может привести к трудностям в разработке крупных проектов.

React

Простой фреймворк, направленный на удобную разработку интерфейсов. Имеет большую поддержку сообществом, однако во время работы над крупным проектом неправильный выбор технологий может привести к проблемам с поддержкой и ошибкам.

Vue.js

Vue.js отличается понятной и рациональной архитектурой, которую несложно освоить и просто применять на практике. Фреймворк предоставляет возможность постепенного внедрения в проект.

Dojo

В Dojo большое внимание уделено модульности компонентов платформы, имеет систему интернационализации и шаблоны для обеспечения доступности приложений. Фреймворк нацелен на предоставление структурированного окружения разработки, удобного для программиста.

Ember

Наиболее строго типизированный распространенный фреймворк. Предоставляет продвинутую систему управления версиями своей платформы, инструменты для перехода на новые версии, и чёткие руководства и средства по обходу устаревших API. Имеет чёткую и рациональную архитектуру, подходящую для разработки сложных веб-приложений.

Aurelia

Имеет хорошую структуру и методику создания веб-приложений, использует современные технологии. Тем не менее фреймворку не хватает более обширного сообщества, способного помочь его разработке и развитию.

UI-библиотека:

Ant Design

Предлагаемый набор включает как базовые, так и расширенные компоненты. Сами компоненты обладают приятным дизайном и содержат множество настроек и опций. Благодаря понятному API и документации библиотека достаточно легка в освоении. Однако большой объем Ant Design может привести к проблемам с производительностью.

Mantine

Новая развивающаяся библиотека с открытым кодом. Обладает универсальностью и предлагает удобные к адаптации под свой проект компоненты.

Material UI

Библиотека позволяет создать адаптивные интерфейсы под разные разрешения экрана, поддерживает все современные браузеры, предоставляет готовые компоненты с настроенным дизайном и позволяет среди них импортировать только необходимые. Тем не менее большой размер библиотеки

снижает скорость загрузки приложения. Material UI довольно сложен для пользования, не всегда успешно интегрируется с другими библиотеками и имеет ограниченную кастомизацию.

Tailwind CSS

Библиотека позволяет создавать сложные адаптивные макеты, имеет собственные служебные шаблоны, обладает легкой настройкой. Однако большое количество классов иногда приводит к трудностям в выборе необходимого.

Bootstrap

Весьма популярная библиотека, позволяющая создавать адаптивные сайты, правильно отображающиеся на разных устройствах и браузерах. Bootstrap легок в освоении и использовании, однако имеет свои ограничения и приводит к созданию шаблонных сайтов, похожих на множество других.

Веб-сервер:

Apache

Apache – это программное обеспечение с открытым исходным кодом, разработанное и поддерживаемое открытым сообществом разработчиков и работающее в самых разных операционных системах. Более прост в использовании, поддерживается всеми операционными системами, предоставляет возможность добавления модулей.

Nginx

Nginx – это программное обеспечение с открытым исходным кодом, которое позволяет создавать веб-сервер. Обеспечивает хорошую производительность для статического и динамического контента, высокую безопасность, имеет хорошую поддержку.

Итоговый выбор:

В качестве фреймворка был выбран удобный и популярный React. Для разработки интерфейса использовалась библиотека Material UI, предлагающая множество готовых настроенных компонентов. Веб-сервером выступает Nginx, предоставляющий такие возможности, как, например, настройку обратного прокси.

Вывод:

Таким образом, был определен предоставляемый пользователю функционал системы, проведен анализ аналогичных проекту сайтов и выбраны инструменты разработки серверной и клиентской части.

2. ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

2.1. Разработка структуры системы

Рассмотрим структуру системы (см. рис. 4):

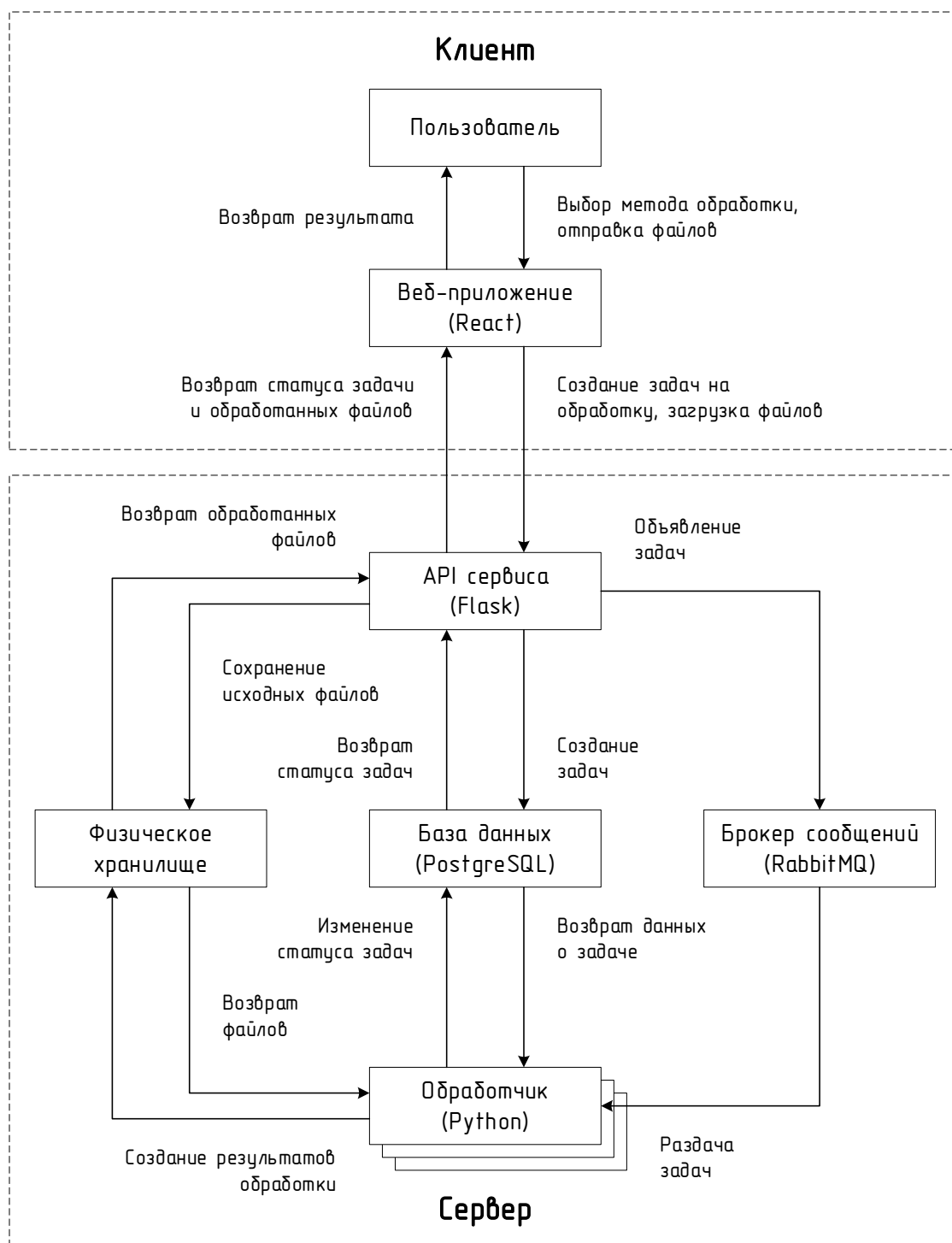


Рис. 4. Структура системы

Взаимодействие между сервером и клиентским приложением предполагается вести через API (см. рис. 4). Обращаясь к нужным методам, клиент отправляет файлы, указывает требуемый тип обработки и соответствующие параметры, проверяет статус обработки и получает готовый результат.

Часть сервера, отвечающая за обработку API, при получении новой задачи сохраняет загруженные файлы на физическом хранилище, создает соответствующую задаче запись в базе данных, где также указывается местоположение исходных файлов, и заносит сообщение о задаче в очередь брокера сообщений.

Чтобы определить статус выполнения задачи, сервер делает обращение в базу данных. Если задача оказывается завершенной, то появляется возможность возвращения готового результата. Местоположение обработанного файла можно получить из записи соответствующей задачи в базе данных.

Предполагается запуск сразу нескольких обработчиков, количество которых можно увеличить при возникновении необходимости в большей мощности работы системы. Изначально обработчики производят прослушивание очереди брокера сообщений. Во время появления новой задачи один из обработчиков забирает данную задачу. Затем через обращение к базе данных определяется тип алгоритма обработки и необходимые параметры, а также местоположение исходного файла. Завершив непосредственную обработку, обработчик сохраняет результат на физическом хранилище и указывает его в базе данных, а затем возвращается к прослушиванию очереди брокера сообщений.

Кроме того, на протяжении всех этапов работы с задачей обработчик изменяет ее статус в базе данных. Изначально все задачи при своем создании получают статус «pending». Когда задача попадает к обработчику, тот обновляет ее статус до «processing». После успешного завершения обработки устанавливается статус «finished». В случае возникновения ошибок обработчик

указывает статус «error», прекращает обработку задачи и возвращается к брокеру сообщений.

Во избежание возникновения конфликтов имен при сохранении файлов на физическом хранилище предполагается выдавать им уникальные имена. Исходные и итоговые имена будут храниться в соответствующей записи в базе данных.

2.2. Структура базы данных

Рассмотрим структуру базы данных (см. рис. 5):

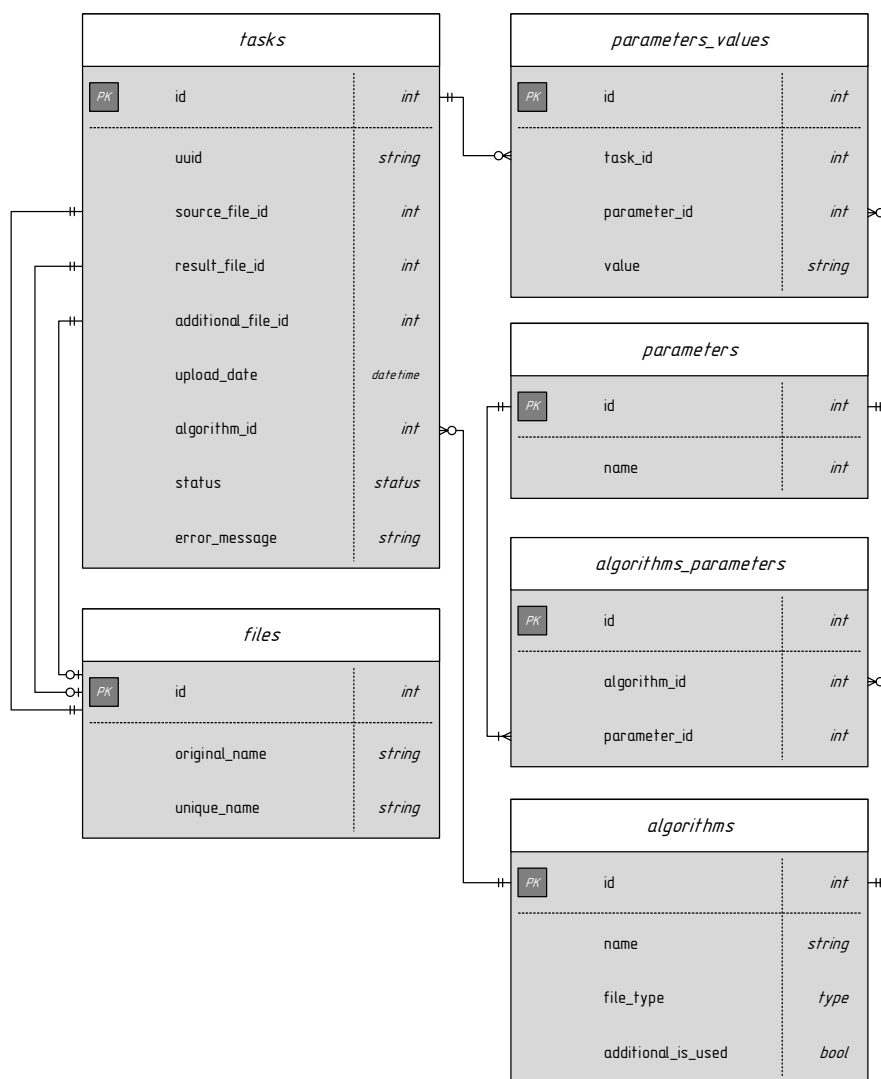


Рис. 5. Структура базы данных

База данных состоит из следующих таблиц (см. рис. 5):

- files – таблица загруженных и готовых обработанных файлов;
- tasks – таблица, хранящая созданные задачи на обработку файлов;
- algorithms – таблица доступных алгоритмов;
- parameters – таблица параметров, необходимых для работы алгоритмов;
- algorithms_parameters – таблица реализации связи «многое-ко-многим» между таблицами algorithms и parameters;
- parameters_values – таблица, устанавливающая соответствие между задачами на обработку и заданными параметрами;
- status – предоставляет этапы обработки задачи:
 - pending – ожидает выполнения,
 - processing – обрабатывается,
 - finished – выполнено,
 - error – закончилось ошибкой;
- type – предоставляет доступные для обработки типа мультимедиа:
 - image – изображения,
 - audio – аудио,
 - video – видео.

Таблицы имеют следующие столбцы (см. табл. 1):

Табл. 1. Столбцы таблиц базы данных

Таблица	Название	Тип	NN	U	PK	FK	Default
files	id	integer	+	+	+		
	Идентификатор						
	original_name	string	+				
	Исходное имя загруженного файла						
	unique_name	string	+				
	Уникальное имя файла, находящегося на физическом хранилище						
tasks	id	integer	+	+	+		
	Идентификатор						
	uuid	string	+	+			
	Уникальный идентификатор, использующийся в формировании ссылок						

	source_file_id	integer	+			files.id	
	Исходный загруженный файл						
	result_file_id	integer				files.id	
	Итоговый обработанный файл						
	additional_file_id	integer				files.id	
	Дополнительный файл, используемый в некоторых алгоритмах						
	upload_date	datetime	+				
	Дата создания задачи						
	algorithm_id	integer	+			algorithms.id	
	Алгоритм обработки файла						
	status	status	+				pending
	Этап выполнения задачи						
	error_message	string					
	Сообщение ошибки, произошедшей во время выполнения алгоритма						
algorithms	id	integer	+	+	+		
	Идентификатор						
	name	string	+				
	Название алгоритма						
	file_type	type	+				
	Тип обрабатываемого файла						
	additional_is_used	bool	+				false
	Необходим ли дополнительный файл для работы алгоритма						
algorithms-parameters	id	integer	+	+	+		
	Идентификатор						
	algorithm_id	integer	+			algorithms.id	
	Алгоритм						
	parameter_id	integer	+			parameters.id	
	Параметр, необходимый для выполнения алгоритма						
parameters	id	integer	+	+	+		
	Идентификатор						
	name	string	+				
	Название параметра						
parameters-values	id	integer	+	+	+		
	Идентификатор						
	task_id	integer	+			tasks.id	
	Задача						
	parameter_id	integer	+			parameters.id	
	Параметр алгоритма						
	value	string	+				
	Значение параметра						

2.3. Описание API

Сервер имеет следующее API (см. табл. 2):

Табл. 2. API сервера

[POST] /processing/process/		
Описание	Создание и запуск новой задачи	
Формат запроса (form)	file	Файл, отданный на обработку
	algorithm	Метод обработки
	parameters	Параметры метода обработки
	additional_file	Дополнительный файл, необходимый для выполнения метода
Формат ответа (JSON)	id	ID созданной задачи
[GET] /processing/status/<id>		
Описание	Получение информации о задаче	
Формат запроса (строка)	id	ID задачи
Формат ответа (JSON)	source name	Название исходного файла
	result name	Название файла результата
	upload date	Дата создания задачи
	type	Тип мультимедиа
	algorithm	Метод обработки файла
	parameters	Параметры метода обработки
	additional name	Название дополнительного файла
	status	Статус выполнения задачи
	error message	Сообщение об ошибке
[GET] /processing/source/<id>		
Описание	Получение исходного файла	
Формат запроса (строка)	id	ID задачи
Формат ответа (BLOB)		Исходного файл
[GET] /processing/result/<id>		
Описание	Получение итогового обработанного файла	
Формат запроса (строка)	id	ID задачи
Формат ответа (BLOB)		Итоговый обработанный файл

[GET] /processing/additional/<id>		
Описание	Получение дополнительного файла	
Формат запроса (строка)	id	ID задачи
Формат ответа (BLOB)		Дополнительный файл

Методы обработки файлов требуют следующие параметры (см. табл. 3):

Табл. 3. Параметры алгоритмов обработки

Название	Тип	Доп.	Параметры	
			Название	Описание
add watermark	image	+	alpha	Непрозрачность водяного знака
compress image	image	-	quality	Качество итогового изображения
convert image	image	-	format	Итоговый формат изображения
crop_image	image	-	left	Координата левого верхнего угла фрагмента по оси X
			upper	Координата левого верхнего угла фрагмента по оси Y
			right	Координата правого нижнего угла фрагмента по оси X
			lower	Координата правого нижнего угла фрагмента по оси Y
flip_image	image	-	method	Направление отражения:
				left right Слева направо
				top bottom Сверху вниз
resize_image	image	-	width	Итоговая ширина изображения
			height	Итоговая высота изображения
			resample	Способ интерполяции:
				nearest Ближайший сосед
				box Бокс-средний
				bilinear Билинейный
				hamming Хэммингово окно
				bicubic Бикубический
				lanczos Ланцош
rotate_image	image	-	angle	Угол поворота против часовой стрелки
			resample	Способ интерполяции:
				nearest Ближайший сосед
				bilinear Билинейный
				bicubic Бикубический
spread_image	image	-	distance	Расстояние рассеивания пикселей
clip_audio	audio	-	start	Время начала фрагмента
			end	Время окончания фрагмента
convert_audio	audio	-	format	Итоговый формат аудио
fade_audio	audio	-	fade in	Длительность затухания в начале
			fade out	Длительность затухания в конце
gain_audio	audio	-	left gain	Изменение громкости в левом канале
			right gain	Изменение громкости в правом канале
join_audio	audio	+	crossfade	Длительность перехода между файлами
reverse_audio	audio	-		
set mono	audio	-		
speed audio	audio	-	multiplier	Коэффициент ускорения аудио
blackwhite_video	video	-		

clip_video	video	-	start	Время начала фрагмента	
			end	Время окончания фрагмента	
convert_video	video	-	format	Итоговый формат видео	
crop_video	video	-	left	Координата левого верхнего угла фрагмента по оси X	
			upper	Координата левого верхнего угла фрагмента по оси Y	
			right	Координата правого нижнего угла фрагмента по оси X	
			lower	Координата правого нижнего угла фрагмента по оси Y	
extract_audio	video	-			
fade_video	video	-	fade_in	Длительность затухания в начале	
			fade_out	Длительность затухания в конце	
flip_video	video	-	method	Направление отражения:	
				left_right	Слева направо
				top_bottom	Сверху вниз
gain_video	video	-	multiplier	Коэффициент усиления громкости	
join_video	video	+	crossfade	Длительность перехода между файлами	
reverse_video	video	-			
rotate_video	video	-	angle	Угол поворота против часовой стрелки	
speed_video	video	-	multiplier	Коэффициент ускорения видео	

Вывод:

Таким образом, была разработана структура системы проекта, ее базы данных, а также описан API, необходимый для взаимодействия серверной и клиентской части.

3. КОНТРОЛЬ КАЧЕСТВА И ИНТЕГРАЦИЯ КОМПОНЕНТОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

3.1. Интерфейс клиентской части

Рассмотрим интерфейс клиентской части:

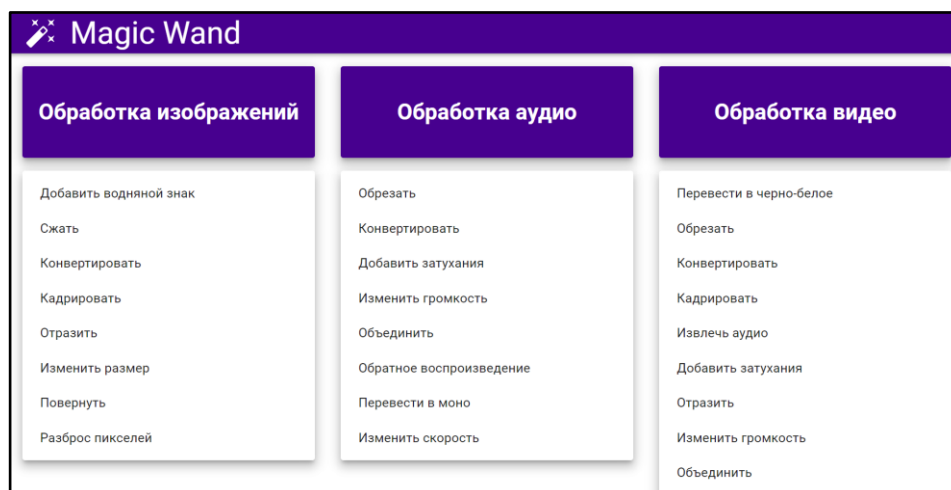


Рис. 6. Главная страница сайта

Главная страница сайта (см. рис. 6) состоит из трех секций, разделенных по типу обрабатываемых файлов: изображения, аудио и видео. Каждая секция содержит внутри себя доступные методы обработки мультимедиа.

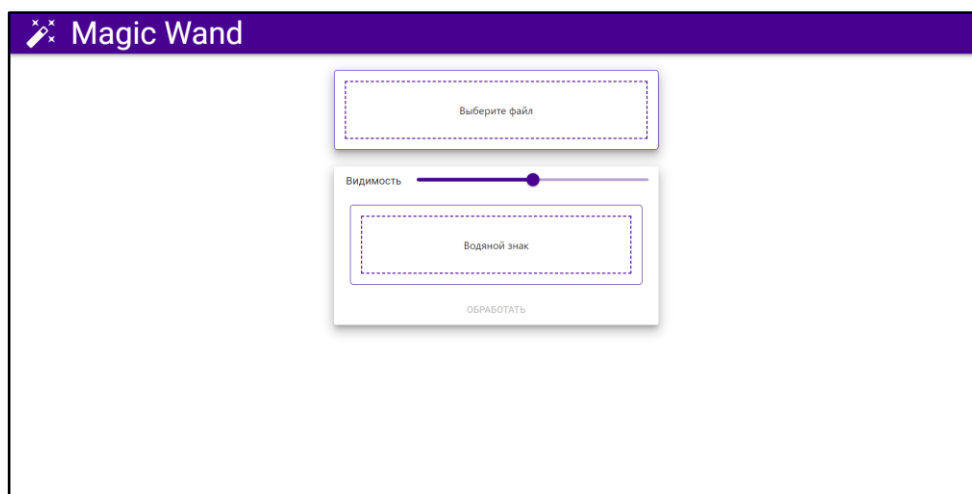


Рис. 7. Пример страницы метода обработки файла

Каждая страница выбранного метода обработки файла (см. рис. 7) имеет поле для загрузки требуемого типа мультимедиа. Различные алгоритмы имеют разный набор параметров, среди которых может встречаться задание числового значения, выбор значения из списка, отправка дополнительного файла. Для некоторых алгоритмов вследствие отсутствия необходимости секция с параметрами может отсутствовать.

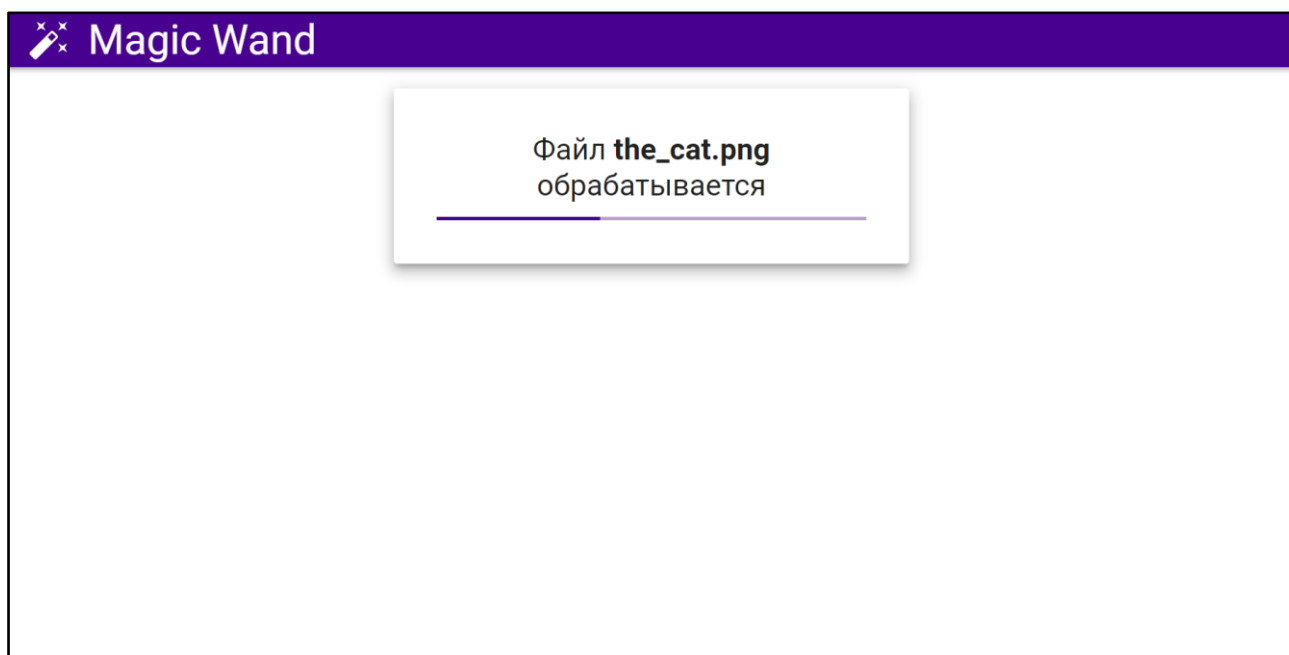


Рис. 8. Пример страницы ожидания окончания обработки

Страница ожидания окончания обработки файла (см. рис. 8) содержит сообщение о том, что выбранный файл в данный момент находится в обработке. Также указывается название отправленного файла.

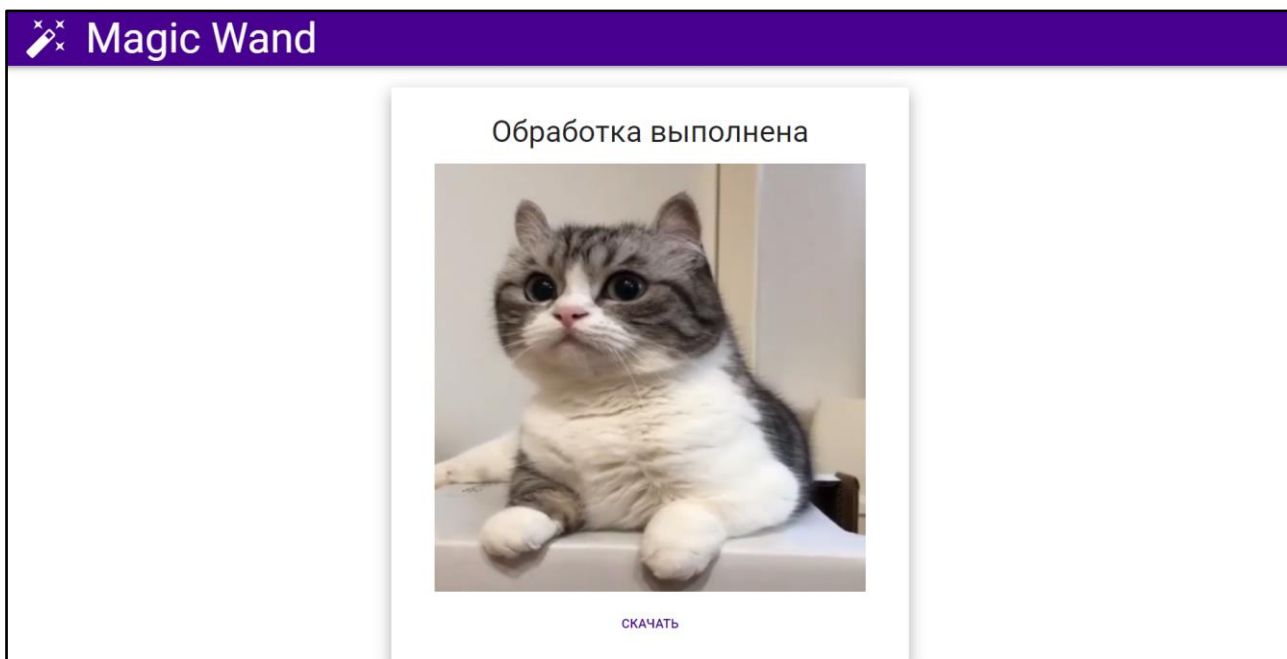


Рис. 9. Страница результата обработки

Страница результата обработки (см. рис. 9) отображает итоговый файл. Данный файл можно скачать с помощью нижерасположенной соответствующей кнопки.

3.2. Руководство пользователя

Общая последовательность взаимодействия с сайтом выглядит следующим образом. На главной странице (см. рис. 6) необходимо выбрать из представленных методов обработки мультимедиа.

После этого произойдет переход на страницу выбора требуемых параметров (см. рис. 7). Пользователь должен отправить исходный файл нужного формата в указанное окно, а затем указать желаемые настройки обработки.

После подтверждения произойдет следующий переход на страницу ожидания обработки (см. рис. 8). Пользователь не обязан держать данную страницу открытой. При необходимости она может быть закрыта и восстановлена по уникальной ссылке.

Когда обработка будет завершена, произойдет автоматический переход на страницу с результатом обработки (см. рис. 9). На ней пользователь может просмотреть итоговый файл и скачать его путем нажатия на соответствующую кнопку.

Каждый метод обработки мультимедиа имеет свой набор параметров:

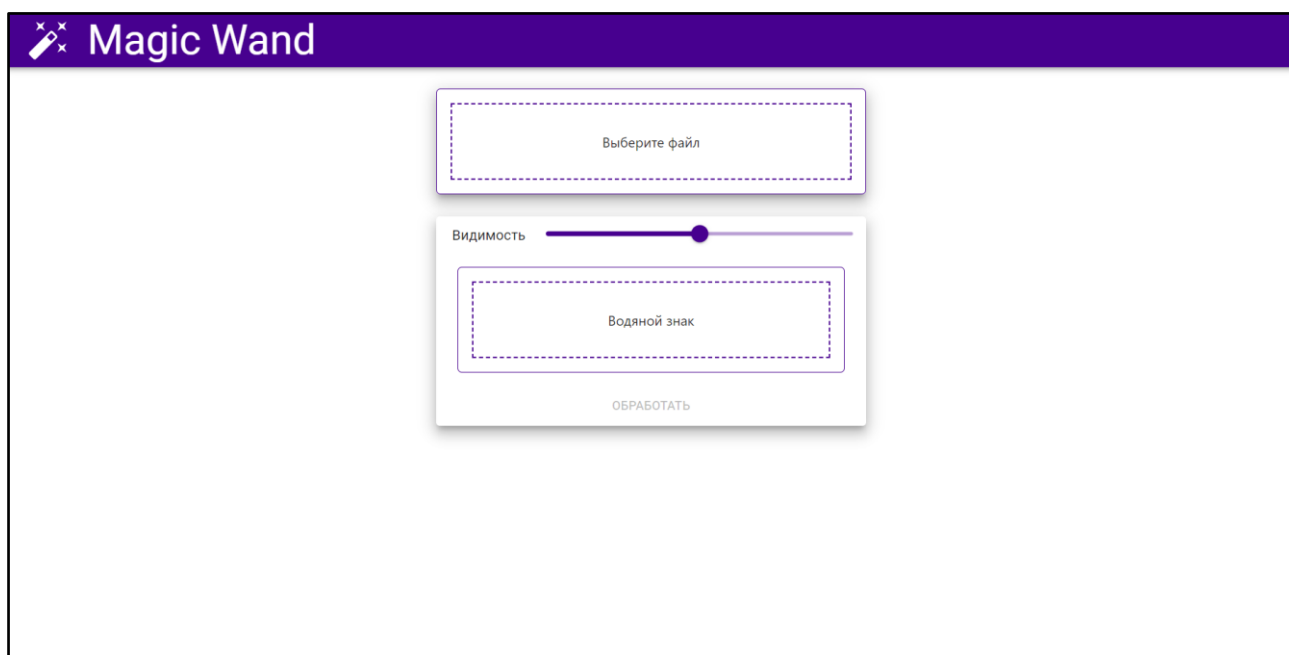


Рис. 10. Добавление водяного знака

Алгоритм добавления водяного знака (см. рис. 10) требует указания степени непрозрачности водяного знака. Данное значение указывается соответствующим ползунком. Крайнее левое положение – ноль – означает, что водяной знак будет полностью прозрачным, то есть не будет виден. Крайнее правое положение – единица – означает полную видимость знака. Сам водяной знак отправляется с помощью нижерасположенной соответствующей секции.

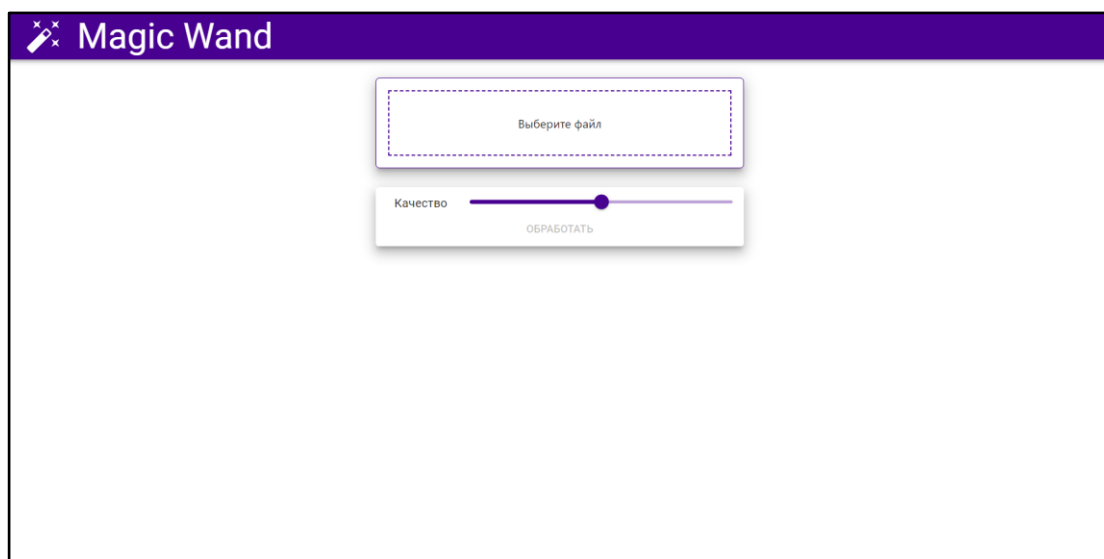


Рис. 11. Сжатие изображения

Алгоритм сжатия изображения (см. рис. 11) требует единственный параметр – число от 0 до 100, задаваемое соответствующим ползунком. Данный параметр влияет на то, каким будет итоговое качество обработанного изображения, например, наилучшим (100%) или наихудшим (0%).

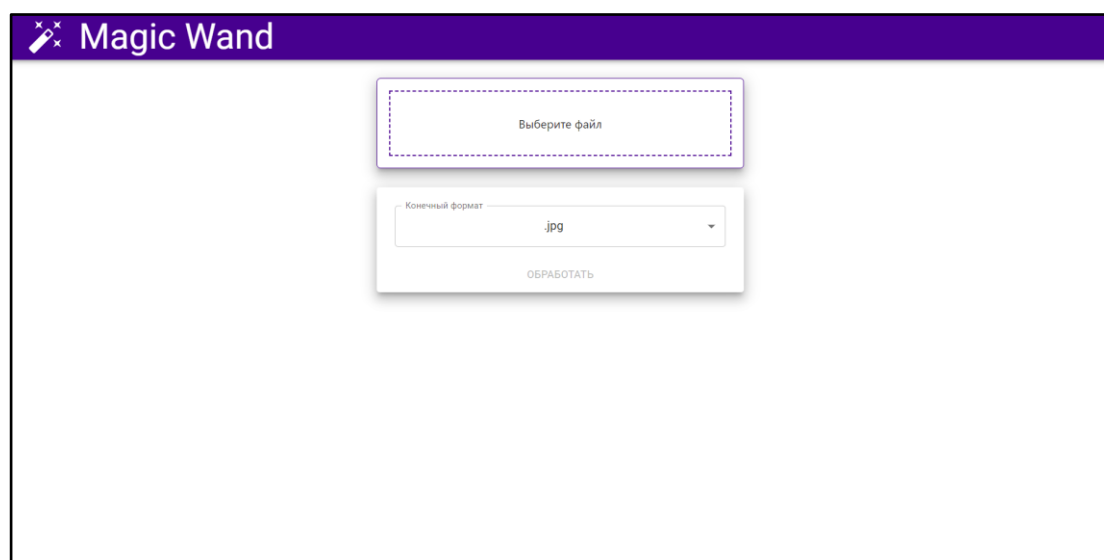


Рис. 12. Конвертация изображения

Для работы данного алгоритма (см. рис. 12) в качестве итогового формата изображения предлагаются на выбор следующие форматы: JPG (JPEG), PNG, BMP, GIF, WEBP.

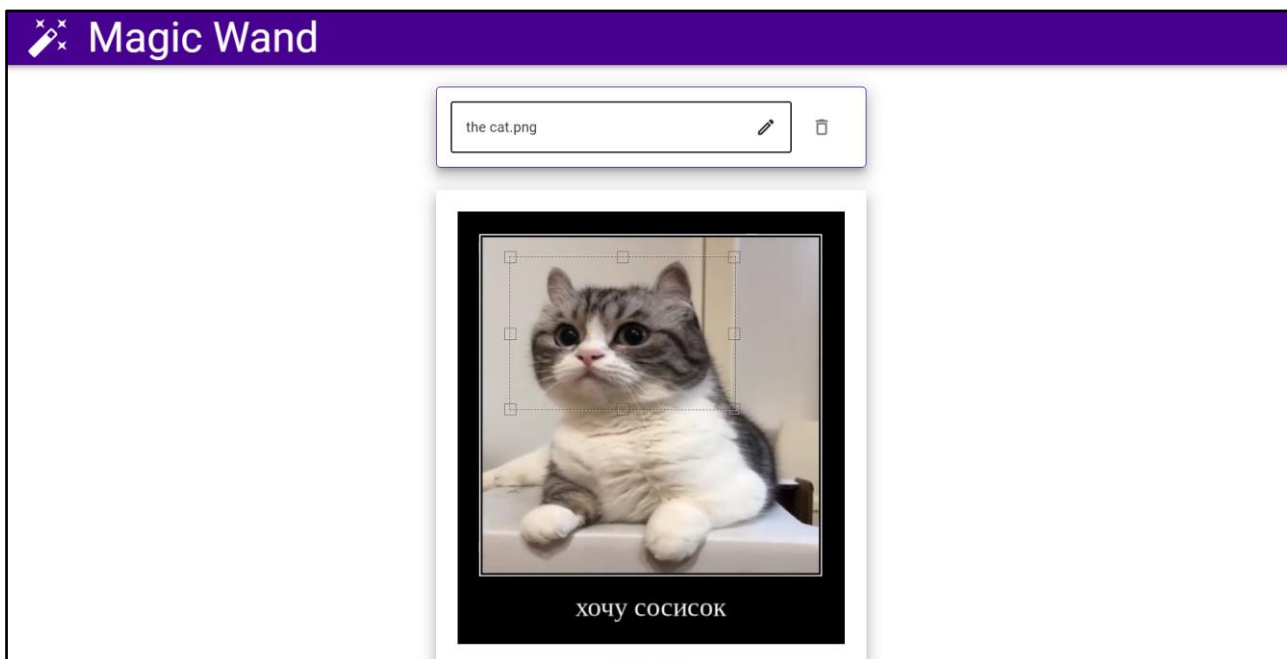


Рис. 13. Кадрирование изображения

Перед тем, как начать задавать параметры алгоритма, сперва следует отправить исходное изображение (см. рис. 13). После этого файл отобразится на сайте и появится возможность выделения требуемого фрагмента.

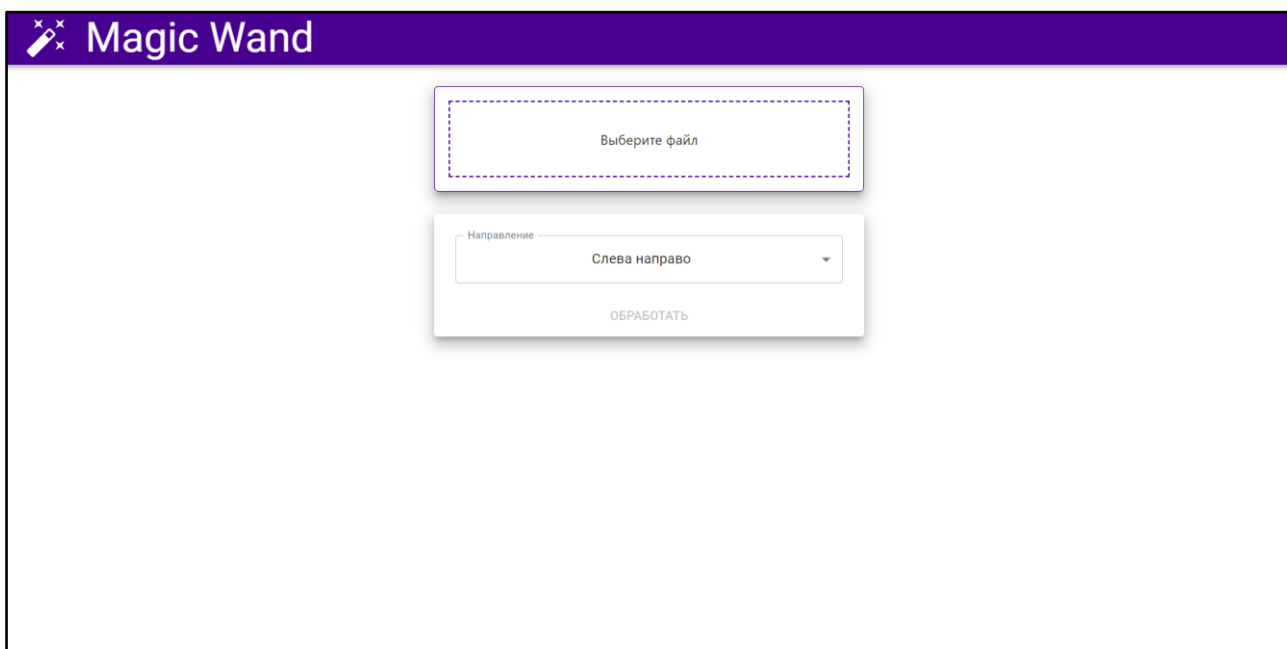


Рис. 14. Отражение изображения

Алгоритм отражения изображения (см. рис. 14) запрашивает один из двух направлений отражения: слева направо или сверху вниз. Аналогичным образом работает соответствующий алгоритм для видео.

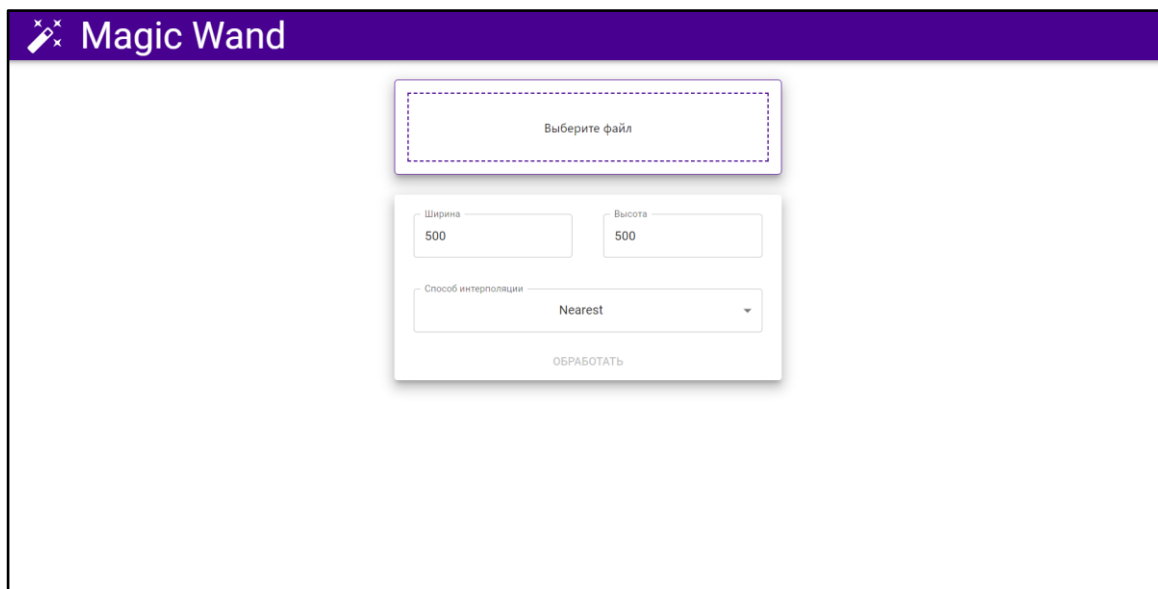


Рис. 15. Изменение размера изображения

Данный алгоритм (см. рис. 15) требует указания новой высоты и ширины изображения, а также способ интерполяции: nearest, box, bilinear, hamming, bicubic, lanczos.

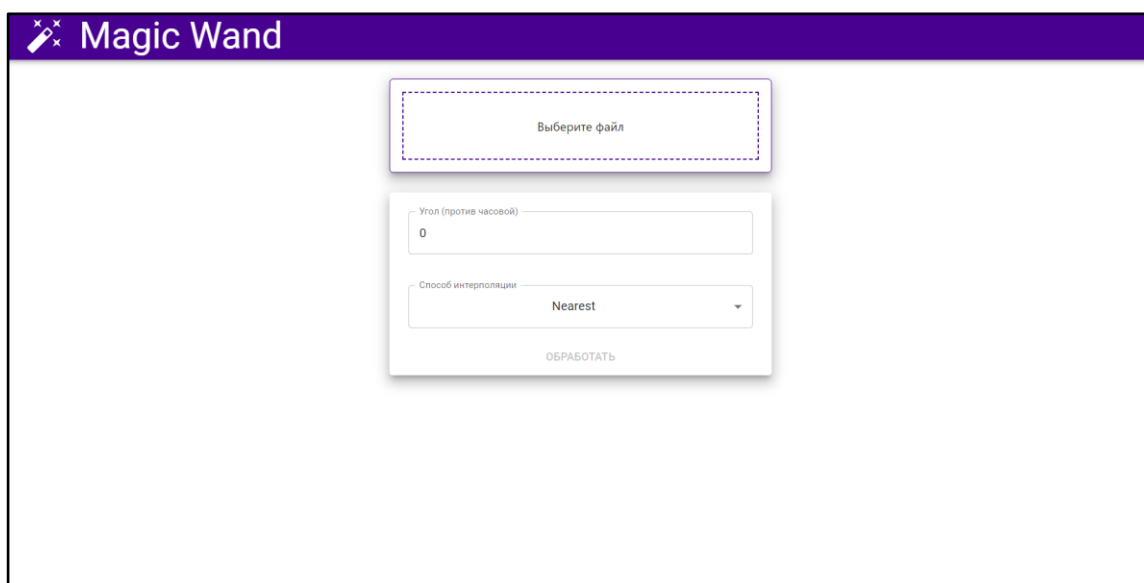
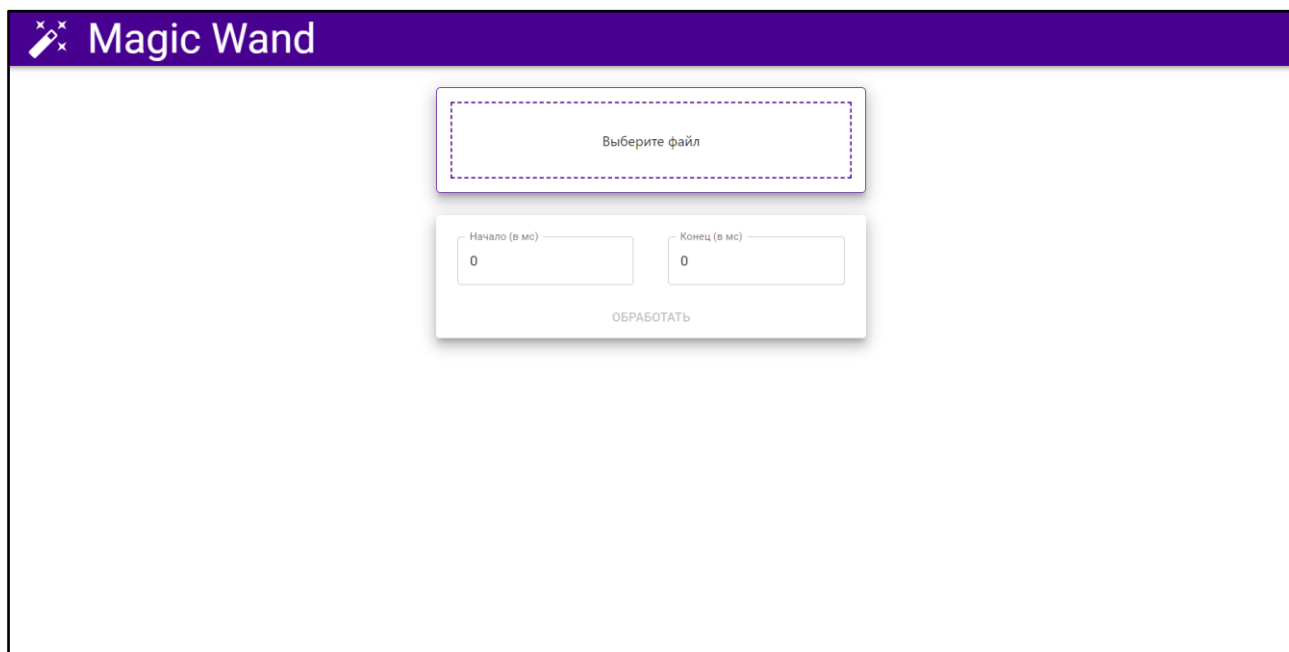


Рис. 16. Поворот изображения

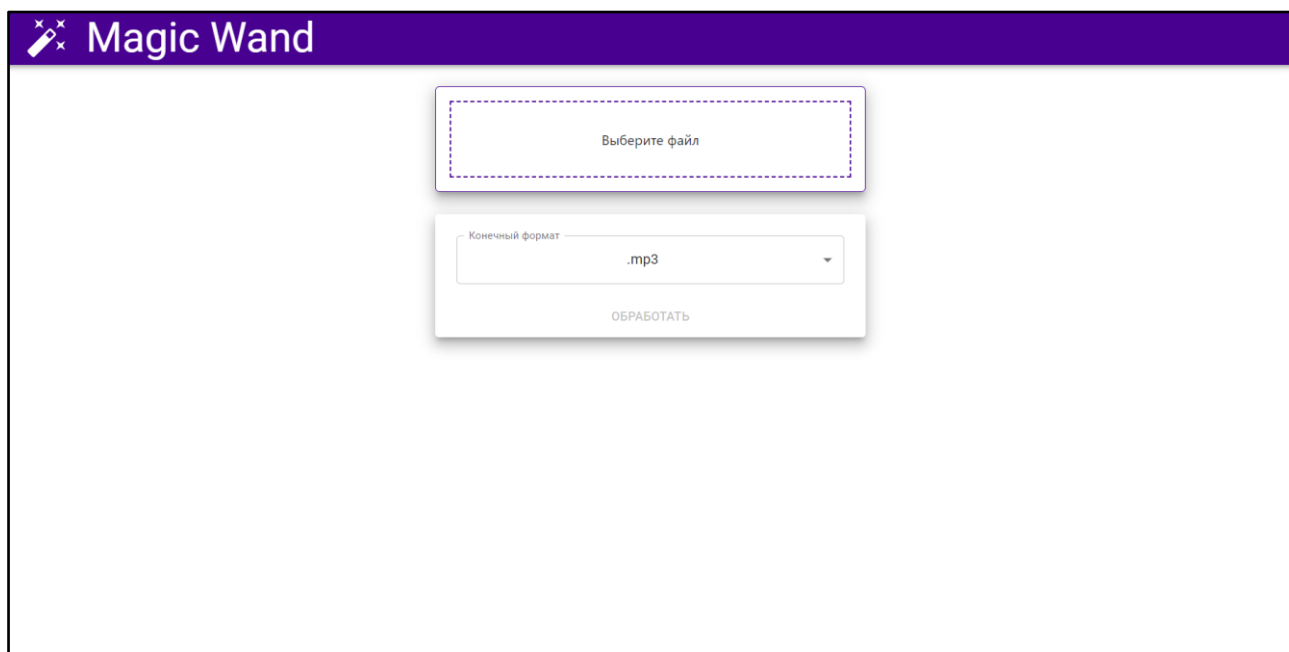
Алгоритм поворота изображения (см. рис. 16) запрашивает угол поворота против часовой стрелки и способ интерполяции: nearest, bilinear, bicubic.



The screenshot shows the 'Magic Wand' application window with a purple header. The main area contains a file selection box with the text 'Выберите файл'. Below it are two input fields: 'Начало (в мс)' with the value '0' and 'Конец (в мс)' with the value '0'. At the bottom of these fields is a button labeled 'ОБРАБОТАТЬ'.

Рис. 17. Извлечение фрагмента из аудио

Алгоритм извлечения фрагмента из аудио (см. рис. 17) требует указания времени начала и окончания нужного фрагмента.



The screenshot shows the 'Magic Wand' application window with a purple header. The main area contains a file selection box with the text 'Выберите файл'. Below it is a dropdown menu labeled 'Конечный формат' with the value '.mp3'. At the bottom of the dropdown is a button labeled 'ОБРАБОТАТЬ'.

Рис. 18. Конвертация аудио

Алгоритм конвертации аудио (см. рис. 18) предлагает следующие форматы итогового файла: MP3, WAV, OGG, FLAC, OPUS, AAC.

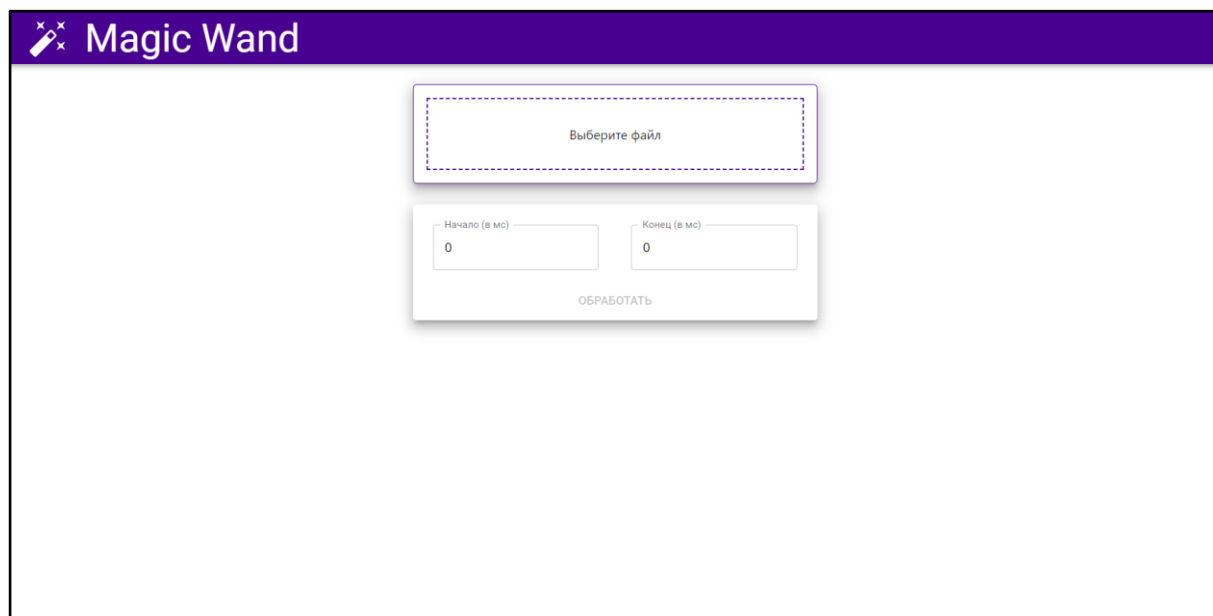


Рис. 19. Добавление затуханий в аудио

Метод добавления затуханий (см. рис. 19) требует указания длительности затухания в начале и в конце. Добавление затуханий в видео происходит аналогичным образом.

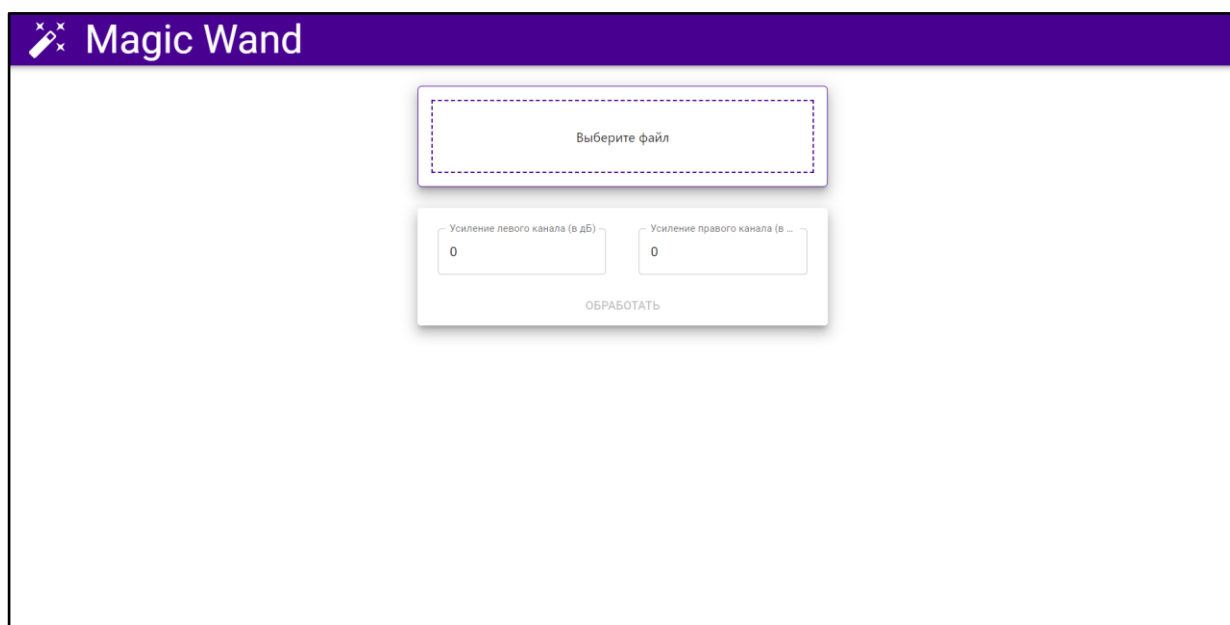


Рис. 20. Изменение громкости аудио

Для работы алгоритма изменения громкости аудио (см. рис. 20) нужно указать, на сколько увеличить (или уменьшить – зависит от знака указанного числа) громкость левого и правого каналов соответственно.

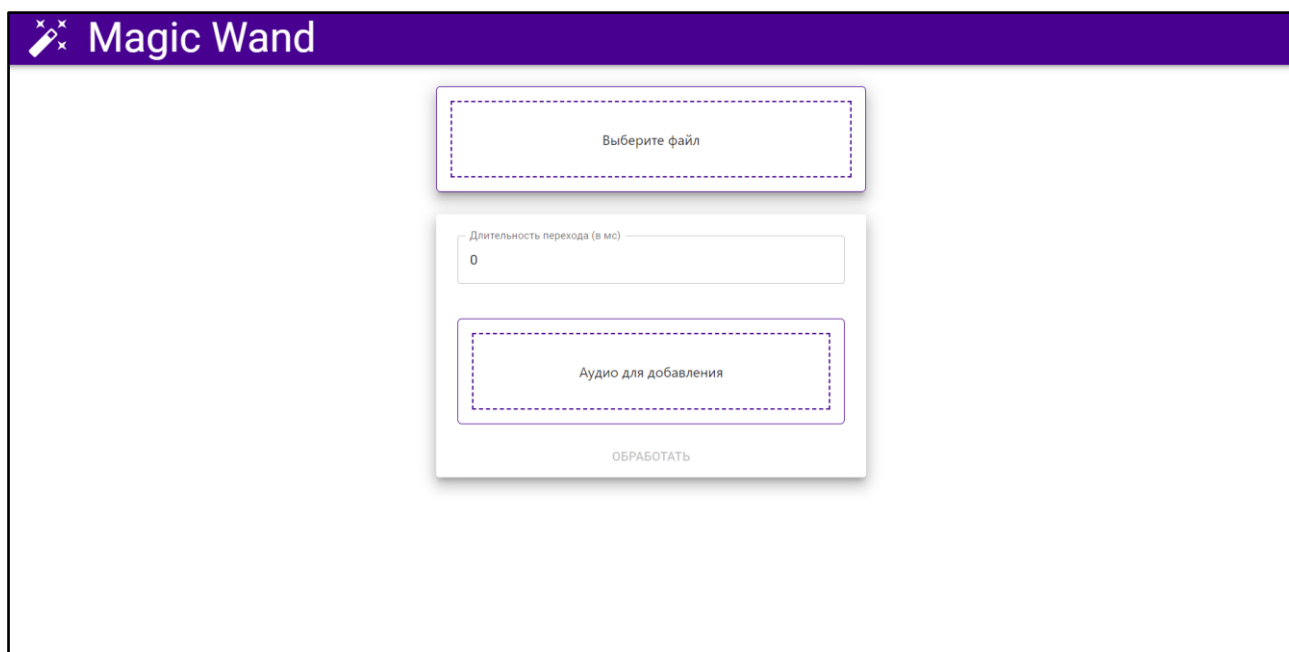


Рис. 21. Объединение аудиофайлов

Метод объединения аудиофайлов (см. рис. 21) запрашивает дополнительный аудиофайл, который будет добавлен после основного, и длительность линейного перехода между ними. Аналогичным образом происходит работа с вариантом алгоритма для видеофайлов.

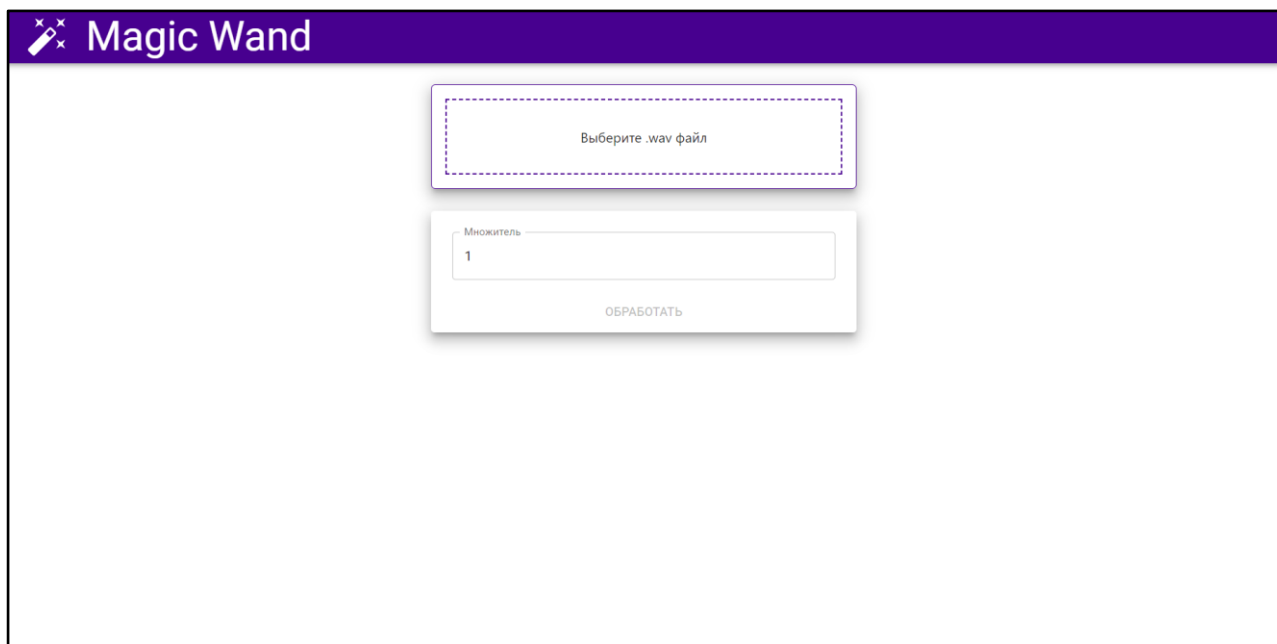


Рис. 22. Изменение скорости аудио

Алгоритм изменения скорости аудио (см. рис. 22) требует указания, во сколько раз следует ускорить (замедлить) воспроизведение файла. Так же происходит работа с аналогичным алгоритмом для обработки видео.

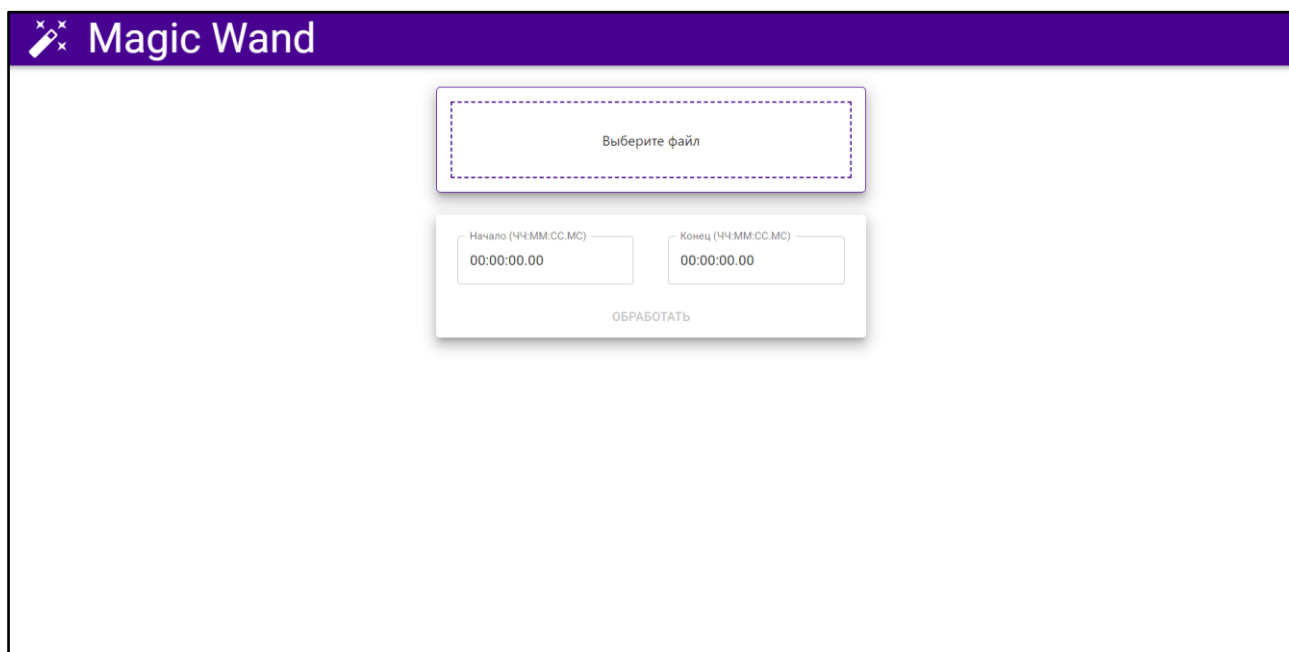


Рис. 23. Извлечение фрагмента из видео

Для работы алгоритмы извлечения фрагмента из видео (см. рис. 23) нужно задать временные точки начала и окончания требуемого фрагмента.

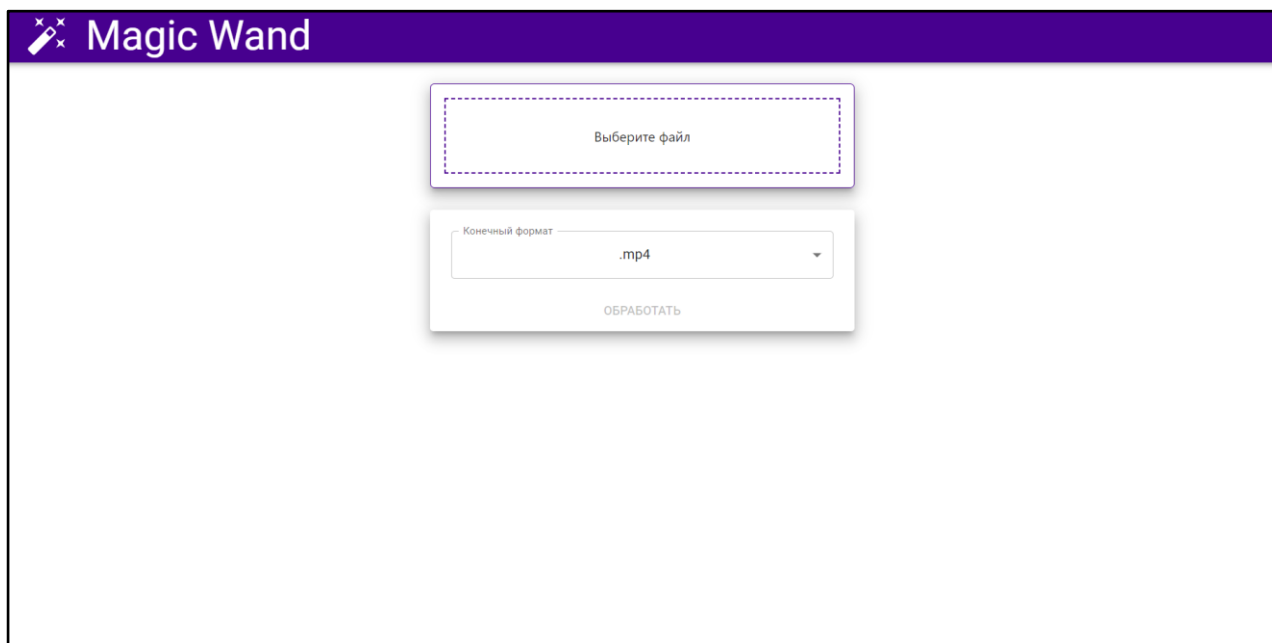


Рис. 24. Конвертация видео

Метод конвертации видео (см. рис. 24) предлагает следующие форматы итогового файла: MP4, AVI, WEBM.

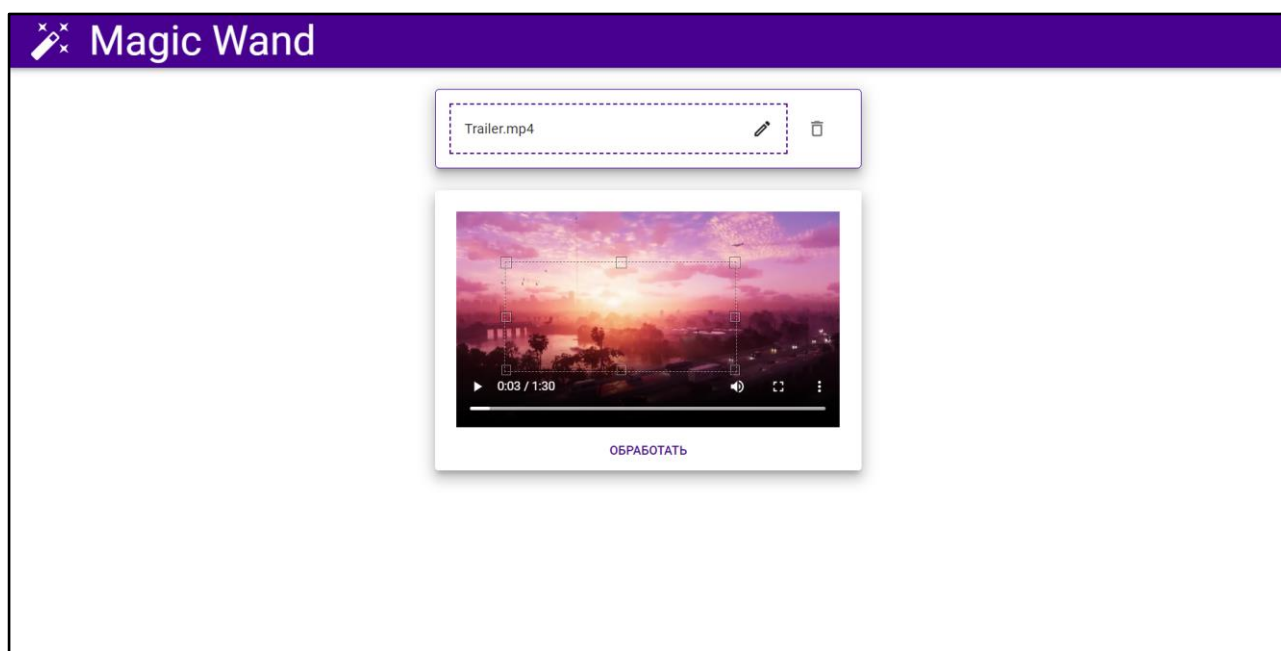
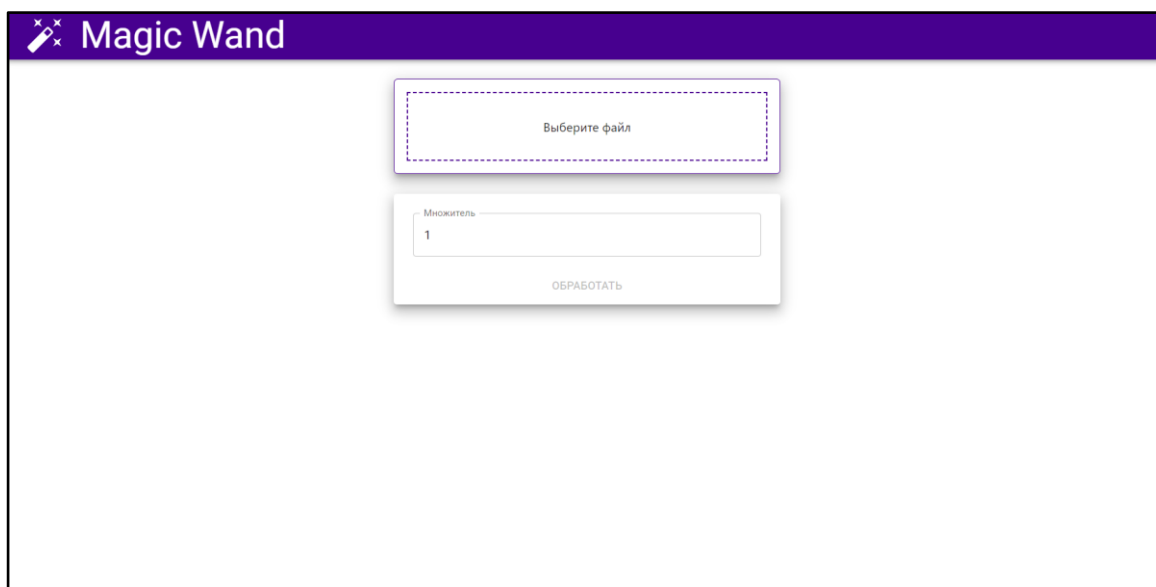


Рис. 25. Кадрирование видео

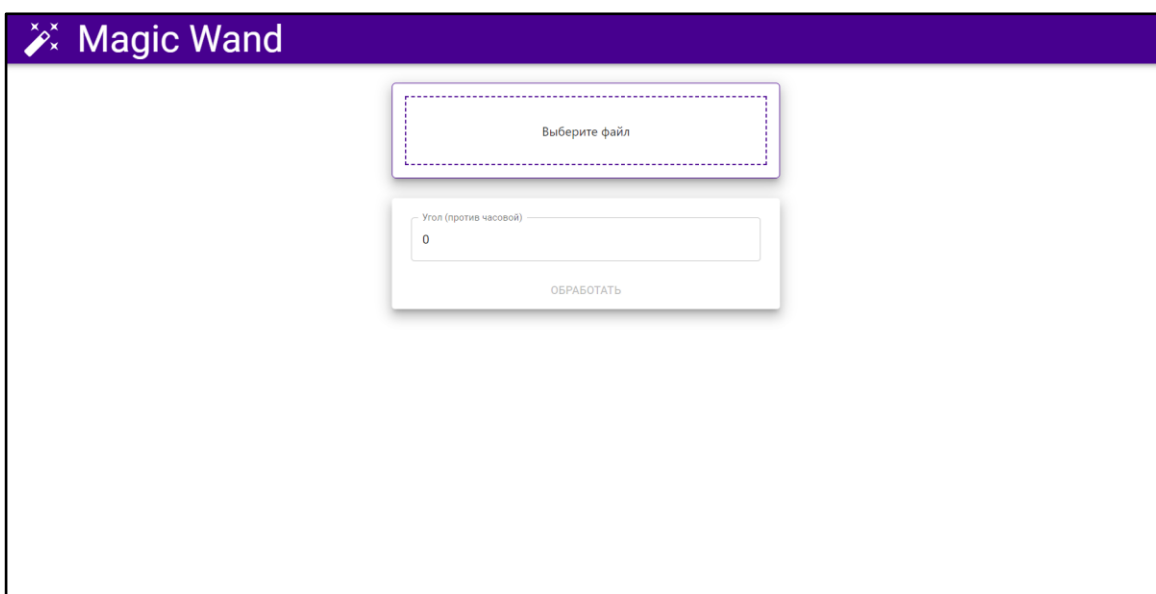
В методе кадрирования видео (см. рис. 25) для открытия доступа к параметрам также необходимо сначала отправить исходный файл. После этого появится возможность выделить на отображенном файле нужный фрагмент.



The screenshot shows the 'Magic Wand' application window. It has a purple title bar with a wand icon and the text 'Magic Wand'. The main area is white and contains two centered UI elements. The top element is a dashed rectangular box with the text 'Выберите файл' (Select file). Below it is a white rectangular box with a light gray border. Inside this box, on the left, is the label 'Множитель' (Multiplier) above a text input field containing the number '1'. At the bottom right of this box is a button labeled 'ОБРАБОТАТЬ' (Process).

Рис. 26. Изменение громкости видео

Алгоритм изменения громкости видео (см. рис. 26) в качестве единственного параметра принимает значение, означающее, во сколько раз сделать громкость видео громче (тише).



The screenshot shows the 'Magic Wand' application window. It has a purple title bar with a wand icon and the text 'Magic Wand'. The main area is white and contains two centered UI elements. The top element is a dashed rectangular box with the text 'Выберите файл' (Select file). Below it is a white rectangular box with a light gray border. Inside this box, on the left, is the label 'Угол (против часовой)' (Angle (counter-clockwise)) above a text input field containing the number '0'. At the bottom right of this box is a button labeled 'ОБРАБОТАТЬ' (Process).

Рис. 27. Поворот видео

Метод поворота видео (см. рис. 27) запрашивает угол поворота против часовой стрелки.

Алгоритмы обратного воспроизведения аудио и видео, перевода аудио в одноканальный звуковой режим, перевода видео в черно-белое отображение и извлечения аудиодорожки из видео не требуют каких-либо параметров. Для их работы достаточно лишь отправить исходный файл и запустить работу метода.

Вывод:

Таким образом, был подробно описан общий принцип взаимодействия с сервисом, а также способы создания задач с различными методами обработки мультимедиа.

ЗАКЛЮЧЕНИЕ

В результате выполнения курсового проекта был определен предоставляемый пользователю функционал системы, проведен анализ аналогичных проекту сайтов и выбраны инструменты разработки серверной и клиентской части, была разработана структура системы проекта, ее базы данных, а также описан API, необходимый для взаимодействия серверной и клиентской части, был подробно описан общий принцип взаимодействия с сервисом, а также способы создания задач с различными методами обработки мультимедиа

Данный сервис удобен для быстрого выполнения простой обработки файлов и освобождает пользователя от необходимости устанавливать на свой ПК дополнительное настольное программное обеспечение.

Сервис можно улучшить путем реализации и интеграции новых алгоритмов обработки файлов, создания клиентов на других платформах, внедрения системы пользователей, открывающей возможность интеграции платных подписок для более продвинутого использования сайта, расширения количества обработчиков для роста производительности.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

Основная литература

1. Базы данных : учебно-методическое пособие / Г. И. Ревунков, Н. А. Ковалева, Е. Ю. Силантьева [и др.]. — Москва : МГТУ им. Н.Э. Баумана, 2020. — 28 с. — ISBN 978-5-7038-5381-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/205187>
2. Годзурас, Э. Docker Compose для разработчика : руководство / Э. Годзурас ; перевод с английского А. Н. Киселева. — Москва : ДМК Пресс, 2023. — 220 с. — ISBN 978-5-93700-203-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/348110>
3. Гринберг, М. Разработка веб-приложений с использованием Flask на языке Python / М. Гринберг. — Москва : ДМК Пресс, 2014. — 272 с. — ISBN 978-5-97060-138-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/90103>
4. де, Й. Д. NGINX. Книга рецептов : руководство / Й. Д. де ; перевод с английского Д. А. Беликова. — Москва : ДМК Пресс, 2020. — 176 с. — ISBN 978-5-97060-790-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/140590>
5. Елисеев, А. И. Разработка веб-приложений с использованием фреймворка Flask : учебное пособие : в 2 частях / А. И. Елисеев, Ю. В. Минин, В. А. Гриднев. — Тамбов : ТГТУ, 2020 — Часть 1 — 2020. — 82 с. — ISBN 978-5-8265-2188-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/320318>
6. Копырин, А. С. Программирование на Python : учебное пособие / А. С. Копырин, Т. Л. Салова. — Москва : ФЛИНТА, 2021. — 48 с. — ISBN 978-

- 5-9765-4753-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/182960>
7. Кузниченко, М. А. Основы баз данных : учебно-методическое пособие / М. А. Кузниченко. — 2-е изд., стер. — Москва : ФЛИНТА, 2022. — 102 с. — ISBN 978-5-9765-5139-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/266339>
 8. Лоре, А. Проектирование веб-API : руководство / А. Лоре ; перевод с английского Д. А. Беликова. — Москва : ДМК Пресс, 2020. — 440 с. — ISBN 978-5-97060-861-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/179498>
 9. Моуэт, Э. Использование Docker / Э. Моуэт ; научный редактор А. А. Маркелов ; перевод с английского А. В. Снастина. — Москва : ДМК Пресс, 2017. — 354 с. — ISBN 978-5-97060-426-7. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/93576>
 10. Осипов, Д. Л. Технологии проектирования баз данных / Д. Л. Осипов. — Москва : ДМК Пресс, 2019. — 498 с. — ISBN 978-5-97060-737-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/131692>
 11. Разработка серверной части веб-ресурса / В. В. Никулин, А. А. Олейников, А. А. Сорокин, А. В. Олейникова. — Санкт-Петербург : Лань, 2023. — 132 с. — ISBN 978-5-507-47868-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/356102>
 12. Рогов, Е. В. PostgreSQL 15 изнутри : руководство / Е. В. Рогов. — Москва : ДМК Пресс, 2023. — 662 с. — ISBN 978-5-93700-178-8. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/348089>
 13. Сейерс, Э. Х. Docker на практике / Э. Х. Сейерс, А. Милл ; перевод с английского Д. А. Беликов. — Москва : ДМК Пресс, 2020. — 516 с. —

- ISBN 978-5-97060-772-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/131719>
14. Сергеев, А.Н. Основы локальных компьютерных сетей [Электронный ресурс]: учебное пособие / А.Н. Сергеев. — Санкт-Петербург : Лань, 2016. — 184 с. — Режим доступа: URL: <https://e.lanbook.com/book/87591>
15. Топорков, С.С. Компьютерные сети для продвинутых пользователей [Электронный ресурс]: учебное пособие / С.С. Топорков. — Москва : ДМК Пресс, 2009. — 192 с. — Режим доступа: URL: <https://e.lanbook.com/book/1170>
16. Хортон, А. Разработка веб-приложений в ReactJS / А. Хортон, Р. Вайс ; перевод с английского Р. Н. Рагимова. — Москва : ДМК Пресс, 2016. — 254 с. — ISBN 978-5-94074-819-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/97339>
17. Янцев, В. В. Web-программирование на Python / В. В. Янцев. — 2-е изд., стер. — Санкт-Петербург : Лань, 2023. — 180 с. — ISBN 978-5-507-46546-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/310289>

Дополнительная литература

18. Ачилов, Р.Н. Построение защищенных корпоративных сетей [Электронный ресурс]: учебное пособие / Р.Н. Ачилов. — Москва : ДМК Пресс, 2013. — 250 с. — Режим доступа: URL: <https://e.lanbook.com/book/66472>
19. Ибе, О. Компьютерные сети и службы удаленного доступа [Электронный ресурс]: справочник / О. Ибе. — Москва : ДМК Пресс, 2007. — 336 с. — Режим доступа: URL: <https://e.lanbook.com/book/1169>