# ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к проекту «Редактор изображений с применением AI-стилизации»

Выполнил: Студент

Проверил: Преподаватель

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ................................................................................................................3

1 ОПИСАНИЕ ПРОЕКТА........................................................................................4

1.1 Техническое задание..................................................................................4

1.2 Функциональность системы......................................................................5

2 ОПИСАНИЕ ИНСТРУМЕНТАРИЯ И ЕГО ОБОСНОВАНИЕ................................7

2.1 Выбор языка программирования..................................................................7

2.2 Выбор среды разработки...........................................................................11

2.3 Выбор библиотек и фреймворков................................................................12

3 ОПИСАНИЕ РЕАЛИЗАЦИИ................................................................................14

3.1 Архитектура приложения............................................................................14

3.2 Описание пользовательского интерфейса...................................................16

ЗАКЛЮЧЕНИЕ.........................................................................................................20

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ........................................................21

# ВВЕДЕНИЕ

В современную эпоху цифровых технологий обработка и стилизация изображений становится всё более востребованной задачей. Пользователи социальных сетей, дизайнеры, маркетологи и обычные люди стремятся делать свои изображения более привлекательными и художественными. Однако профессиональные навыки обработки изображений требуют значительного времени для освоения, а коммерческие программы для редактирования фото часто сложны в использовании и имеют высокую стоимость.

Проект «Редактор изображений с применением AI-стилизации» призван решить эту проблему, предоставляя пользователям простой и интуитивно понятный инструмент для художественной стилизации изображений с использованием технологий искусственного интеллекта. Искусственный интеллект берет на себя сложную работу по анализу и трансформации изображений, позволяя пользователям без специальных навыков получать высококачественные стилизованные результаты.

Данный проект представляет собой кроссплатформенное веб-приложение, которое позволяет загружать изображения, применять к ним различные художественные стили, и сохранять или делиться результатами. Интерфейс приложения разработан с учетом современных стандартов UX/UI дизайна, обеспечивая максимальное удобство для пользователей с разным уровнем подготовки.

Актуальность проекта обусловлена растущим спросом на инструменты обработки изображений в условиях цифровизации коммуникаций и развития социальных сетей, где визуальный контент играет ключевую роль в привлечении внимания аудитории. Использование технологий AI позволяет существенно упростить процесс создания привлекательного визуального контента и сделать его доступным для широкой аудитории.

# 1 ОПИСАНИЕ ПРОЕКТА

## 1.1 Техническое задание

Целью данного проекта является разработка веб-приложения для стилизации изображений с использованием технологий искусственного интеллекта, обладающего следующими характеристиками:

* Интуитивно понятный пользовательский интерфейс, доступный для пользователей без специальных навыков в области дизайна и обработки изображений;
* Возможность загрузки изображений в различных форматах (JPEG, PNG, GIF, WEBP);
* Набор готовых художественных стилей, вдохновленных работами известных художников (Ван Гог, Пикассо, Кандинский и др.);
* Возможность сохранения и экспорта обработанных изображений в различных форматах с сохранением высокого качества;
* Возможность быстрого доступа к ранее обработанным изображениям;
* Поддержка русского языка интерфейса;
* Кроссплатформенность и адаптивный дизайн для корректного отображения на различных устройствах.

Технические требования к аппаратному и программному обеспечению:

* Функционирование в современных браузерах (Chrome, Firefox, Safari, Edge);
* Серверная часть с достаточной производительностью для обработки изображений в реальном времени;
* Оптимизация для работы на устройствах с различной производительностью;
* Отказоустойчивость и защита от потери данных.

## 1.2 Функциональность системы

Разрабатываемое приложение для редактирования изображений предоставляет следующую функциональность:

**1. Управление изображениями:**

* Загрузка изображений с локального устройства через интерфейс drag-and-drop или стандартный диалог выбора файла;
* Просмотр и предварительная обработка изображений перед применением стилей;
* Сохранение обработанных изображений в различных форматах с оптимизацией качества;
* Возможность поделиться обработанными изображениями напрямую в социальных сетях.

**2. Применение AI-стилей:**

* Выбор из предустановленных художественных стилей, основанных на работах известных художников;
* Предварительный просмотр эффекта применения стиля перед финальной обработкой;
* Быстрое переключение между различными стилями для сравнения результатов;
* Сохранение оригинального изображения с возможностью возврата к нему в любой момент.

**3. Дополнительные функции редактирования:**

* Добавление текста с различными шрифтами и стилями;
* Базовая коррекция яркости, контрастности и насыщенности;
* Создание простых коллажей из нескольких изображений;
* Инструменты для кадрирования и изменения размера изображений.

**4. Управление проектами:**

* Автоматическое сохранение истории редактирования для возможности отмены действий;
* Организация обработанных изображений в виде галереи проектов;
* Возможность возвращения к предыдущим проектам для продолжения работы.

# 2 ОПИСАНИЕ ИНСТРУМЕНТАРИЯ И ЕГО ОБОСНОВАНИЕ

## 2.1 Выбор языка программирования

Для разработки веб-приложения «Редактор изображений с применением AI-стилизации» был выбран язык программирования TypeScript для фронтенда, JavaScript (Node.js) для бэкенда и Python для реализации алгоритмов стилизации изображений с использованием искусственного интеллекта.

**TypeScript (фронтенд):**

* Статическая типизация. TypeScript добавляет строгую типизацию в JavaScript, что позволяет обнаруживать многие ошибки на этапе компиляции, а не во время выполнения. Это особенно важно для проекта с большим количеством данных и сложной логикой обработки изображений.
* Поддержка современных возможностей JavaScript. TypeScript включает все возможности современного JavaScript, одновременно обеспечивая совместимость с более старыми браузерами после компиляции.
* Улучшенная поддержка IDE и инструментов разработки. Благодаря статической типизации, IDE могут предоставлять более точное автодополнение и подсказки, что ускоряет процесс разработки и упрощает рефакторинг.
* Высокая совместимость с популярными фреймворками для фронтенд-разработки, такими как React, которые используются в проекте.

**JavaScript/Node.js (бэкенд):**

* Асинхронное программирование. Node.js отлично подходит для обработки многочисленных одновременных запросов, что идеально для веб-приложения с обработкой изображений, где несколько пользователей могут одновременно загружать и обрабатывать файлы.
* Единый язык для фронтенд и бэкенд разработки. Использование JavaScript/TypeScript на обеих сторонах позволяет повторно использовать код и типы, упрощает коммуникацию между клиентской и серверной частями приложения.
* Богатая экосистема пакетов NPM, включая библиотеки для обработки изображений, которые можно использовать для базовых операций редактирования.
* Хорошая интеграция с другими языками и технологиями через дочерние процессы, что позволяет легко интегрировать Python для AI-стилизации.

**Python (AI-обработка):**

* Богатая экосистема библиотек для машинного обучения и обработки изображений. Python является стандартом для разработки AI-приложений благодаря таким библиотекам как TensorFlow, PyTorch и библиотекам компьютерного зрения, как OpenCV.
* Простота интеграции с предварительно обученными моделями. Python позволяет легко интегрировать готовые модели для трансформации стиля изображений, такие как модели из TensorFlow Hub.
* Высокая производительность для научных вычислений. Библиотеки для работы с массивами данных, такие как NumPy, оптимизированы для работы с изображениями и матричными операциями, необходимыми для AI-стилизации.
* Поддержка модели Google Magenta для трансформации стиля изображений, которая является одной из наиболее эффективных для задач художественной стилизации.

## 2.2 Выбор среды разработки

Для разработки проекта были выбраны следующие среды разработки и инструменты:

**Visual Studio Code:**

* Кросс-платформенность. VS Code доступен на всех основных операционных системах, что обеспечивает согласованность среды разработки для всех членов команды.
* Поддержка TypeScript. Встроенная поддержка TypeScript с функциями автодополнения, проверки типов и подсветки синтаксиса.
* Расширяемость. Богатая экосистема расширений для поддержки различных языков программирования, инструментов и фреймворков, используемых в проекте.
* Встроенные средства отладки для Node.js и возможность установки расширений для отладки Python.
* Интеграция с Git для контроля версий, что упрощает совместную работу над проектом.

**PyCharm (для разработки модуля AI-стилизации):**

* Специализированная среда разработки для Python с расширенной поддержкой для научных вычислений и машинного обучения.
* Интеграция с популярными библиотеками машинного обучения, включая TensorFlow и PyTorch.
* Удобные инструменты для отладки и профилирования Python-кода, что критически важно для оптимизации производительности алгоритмов стилизации.
* Поддержка виртуальных окружений Python для изоляции зависимостей проекта.

## 2.3 Выбор библиотек и фреймворков

Для реализации проекта были выбраны следующие библиотеки и фреймворки:

**Фронтенд:**

* React - JavaScript-библиотека для создания пользовательских интерфейсов. Компонентный подход React позволяет создавать сложные интерфейсы из небольших изолированных частей кода.
* TailwindCSS - утилитарный CSS-фреймворк, который позволяет быстро создавать адаптивные и кастомизированные интерфейсы без написания большого количества CSS-кода.
* ShadcnUI - набор компонентов пользовательского интерфейса, основанный на Radix UI и стилизованный с помощью TailwindCSS, обеспечивающий доступные и легко настраиваемые UI-компоненты.
* React-Dropzone - библиотека для удобной загрузки файлов методом drag-and-drop.
* React-Hook-Form - библиотека для управления формами с поддержкой валидации и оптимизацией производительности.
* TanStack Query (React Query) - библиотека для управления серверным состоянием, кэширования данных и асинхронных запросов.

**Бэкенд:**

* Express - минималистичный и гибкий веб-фреймворк для Node.js, предоставляющий надежную функциональность для веб-приложений.
* Drizzle ORM - современная ORM-библиотека для TypeScript с поддержкой типизации и миграций баз данных.
* Sharp - высокопроизводительная библиотека для обработки изображений в Node.js.
* Zod - библиотека для валидации схем данных с поддержкой типизации TypeScript.

**AI-стилизация:**

* TensorFlow - открытая программная библиотека для машинного обучения, разработанная Google, используемая для создания и обучения моделей глубокого обучения.
* TensorFlow Hub - библиотека для повторного использования предварительно обученных моделей. В проекте используется модель arbitrary-image-stylization-v1-256 из Google Magenta для трансформации стиля изображений.
* NumPy - библиотека для научных вычислений с Python, предоставляющая эффективные структуры данных для работы с многомерными массивами и матрицами.
* Pillow (PIL) - библиотека для работы с изображениями в Python, используемая для загрузки, изменения размера и сохранения изображений с высоким качеством.

# 3 ОПИСАНИЕ РЕАЛИЗАЦИИ

## 3.1 Архитектура приложения

Приложение «Редактор изображений с применением AI-стилизации» построено на основе клиент-серверной архитектуры с распределением ответственности между фронтендом, бэкендом и AI-модулем стилизации. Общая архитектура приложения представлена следующими компонентами:

**Клиентская часть (фронтенд):**

* Компоненты пользовательского интерфейса, построенные с использованием React и ShadcnUI.
* Модуль загрузки изображений с использованием React-Dropzone для обработки пользовательских файлов.
* Модуль предварительного просмотра и редактирования изображений с использованием Canvas API для манипуляций с изображениями на стороне клиента.
* Модуль выбора и предварительного просмотра стилей с возможностью быстрого переключения между различными вариантами.
* Сервисный слой для коммуникации с API бэкенда, реализованный с использованием TanStack Query для управления асинхронными запросами и кэширования данных.

**Серверная часть (бэкенд):**

* API-сервер на базе Express.js для обработки запросов от клиента и маршрутизации.
* Модуль управления файлами для временного хранения загруженных и обработанных изображений.
* Слой хранения данных с использованием Drizzle ORM для управления информацией о шаблонах, стилях и сохраненных пользовательских проектах.
* Модуль интеграции с Python-скриптами для AI-стилизации изображений, реализованный через межпроцессное взаимодействие.

**Модуль AI-стилизации (Python):**

* Скрипт загрузки и обработки изображений с использованием библиотеки PIL/Pillow для базовых манипуляций с изображениями.
* Интеграция с TensorFlow и TensorFlow Hub для загрузки предварительно обученной модели для трансформации стиля.
* Модуль применения стиля с использованием модели Google Magenta для переноса стиля с одного изображения на другое.
* Модуль оптимизации качества изображения для сохранения высокого разрешения при стилизации.

## 3.2 Описание пользовательского интерфейса

Пользовательский интерфейс приложения «Редактор изображений с применением AI-стилизации» разработан с учетом принципов UX/UI дизайна и направлен на обеспечение интуитивно понятного взаимодействия пользователя с системой. Интерфейс состоит из следующих основных экранов и компонентов:

**Главная страница:**

* Приветственный баннер с кратким описанием возможностей приложения и призывом к действию.
* Карточки с основными функциями приложения (Редактор изображений, Генератор мемов, Создание коллажей) с краткими описаниями и кнопками перехода.
* Галерея примеров работ, созданных с помощью приложения, демонстрирующая возможности AI-стилизации.
* Секция с шаблонами для быстрого старта работы.

**Страница редактора изображений:**

* Вкладки «Загрузка» и «Редактирование» для разделения процесса работы на логические этапы.
* Область загрузки с поддержкой drag-and-drop для удобной загрузки изображений.
* Основная рабочая область с предпросмотром изображения и панелью инструментов для базового редактирования.
* Панель выбора стилей с превью эффектов и возможностью предпросмотра результата применения каждого стиля.
* Кнопки для применения выбранного стиля, возврата к оригинальному изображению, сохранения результата и публикации в социальных сетях.
* Индикатор прогресса обработки для информирования пользователя о статусе процесса стилизации.

**Технические особенности интерфейса:**

* Адаптивный дизайн, обеспечивающий корректное отображение на устройствах с различными размерами экрана.
* Темная цветовая схема для основных элементов интерфейса (навигационная панель, карточки) для снижения утомляемости при длительной работе.
* Высококонтрастное оформление текста для обеспечения читаемости (белый текст на темном фоне, темный текст на светлом фоне).
* Интерактивные элементы с визуальной обратной связью (hover-эффекты, анимации переходов) для улучшения пользовательского опыта.
* Система уведомлений для информирования пользователя о результатах операций и возможных ошибках.
* Полностью русифицированный интерфейс с учетом особенностей локализации.

# ЗАКЛЮЧЕНИЕ

В результате выполнения проекта было разработано веб-приложение «Редактор изображений с применением AI-стилизации», которое позволяет пользователям без специальной подготовки создавать художественно стилизованные изображения с использованием технологий искусственного интеллекта.

В ходе работы над проектом были решены следующие задачи:

* Разработан интуитивно понятный пользовательский интерфейс с адаптивным дизайном, обеспечивающий удобство использования на различных устройствах.
* Реализована система загрузки и обработки изображений с поддержкой различных форматов и сохранением высокого качества.
* Интегрирована технология AI-стилизации на основе модели Google Magenta для художественной обработки изображений.
* Создан набор предустановленных стилей, вдохновленных работами известных художников, с возможностью предварительного просмотра и сравнения результатов.
* Реализованы дополнительные функции редактирования, включая добавление текста, базовую коррекцию параметров изображения и создание коллажей.
* Обеспечена возможность сохранения результатов, публикации в социальных сетях и организации работы в виде проектов.

Основные технические особенности и преимущества разработанного приложения:

* Кроссплатформенность и доступность через веб-браузер без необходимости установки дополнительного программного обеспечения.
* Высокая производительность обработки изображений благодаря оптимизации алгоритмов и использованию современных библиотек.
* Сохранение высокого качества и разрешения изображений при стилизации.
* Полностью русифицированный интерфейс с учетом особенностей локализации.

Разработанное приложение может найти применение в различных областях, включая социальные медиа, дизайн, маркетинг, образование и личное творчество. Оно предоставляет пользователям без специальных навыков доступ к передовым технологиям обработки изображений, расширяя возможности для самовыражения и креативных экспериментов.

В перспективе планируется дальнейшее развитие проекта с добавлением новых художественных стилей, расширением функциональности редактирования, интеграцией дополнительных AI-технологий для улучшения качества изображений и оптимизацией производительности обработки для еще более быстрой работы с высокоразрешающими изображениями.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Goodfellow I., Bengio Y., Courville A. Deep Learning. — MIT Press, 2016. — 800 p.

2. Gatys L. A., Ecker A. S., Bethge M. Image Style Transfer Using Convolutional Neural Networks // Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. — 2016. — P. 2414-2423.

3. React Documentation [Электронный ресурс]. — URL: https://reactjs.org/docs/getting-started.html (дата обращения: 30.03.2025).

4. TypeScript Documentation [Электронный ресурс]. — URL: https://www.typescriptlang.org/docs/ (дата обращения: 30.03.2025).

5. TensorFlow Hub [Электронный ресурс]. — URL: https://www.tensorflow.org/hub (дата обращения: 30.03.2025).

6. Google Magenta: Arbitrary Image Stylization [Электронный ресурс]. — URL: https://tfhub.dev/google/magenta/arbitrary-image-stylization-v1-256/2 (дата обращения: 30.03.2025).

7. Express.js Documentation [Электронный ресурс]. — URL: https://expressjs.com/ru/ (дата обращения: 30.03.2025).

8. TailwindCSS Documentation [Электронный ресурс]. — URL: https://tailwindcss.com/docs (дата обращения: 30.03.2025).

9. Sharp Documentation [Электронный ресурс]. — URL: https://sharp.pixelplumbing.com/ (дата обращения: 30.03.2025).

10. Pillow (PIL) Documentation [Электронный ресурс]. — URL: https://pillow.readthedocs.io/ (дата обращения: 30.03.2025).