2 Описание инструментария и его обоснования

2.1 Выбор языка программирования

В наше время существует множество различных языков программирования, и перед созданием приложения программист должен проанализировать, какой язык подойдет для реализации функционала приложения. Для проекта редактора изображений с AI-стилизацией были рассмотрены следующие языки и технологии:

TypeScript и JavaScript - языки программирования, используемые для создания веб-приложений. TypeScript является надмножеством JavaScript, предоставляющим дополнительные возможности статической типизации, что делает разработку крупных приложений более безопасной и структурированной.

Особенности языков программирования:

TypeScript обеспечивает статическую типизацию, что позволяет обнаруживать ошибки на этапе компиляции, а не во время выполнения.

JavaScript является основным языком для веб-разработки, что обеспечивает интеграцию с большинством современных веб-технологий.

Node.js - серверная платформа для JavaScript, которая позволяет создавать быстрые и масштабируемые сетевые приложения. Она использует событийно-ориентированную архитектуру и асинхронный ввод/вывод для оптимизации пропускной способности и масштабируемости.

Python - высокоуровневый интерпретируемый язык программирования с простым синтаксисом, широко используемый в машинном обучении и обработке изображений благодаря богатой экосистеме библиотек для AI и компьютерного зрения.

Особенности языка программирования:

Низкий порог вхождения. Синтаксис Python понятный и читаемый, что упрощает разработку и поддержку кода.

Богатая экосистема библиотек для машинного обучения и обработки изображений, таких как TensorFlow, PyTorch, PIL, NumPy и др.

Проведя анализ популярных языков программирования, для реализации проекта редактора изображений с AI-стилизацией был выбран следующий стек технологий:

1. TypeScript и React для фронтенд-части приложения, что обеспечивает типобезопасность, современную архитектуру и высокую производительность пользовательского интерфейса.

2. Node.js и Express для серверной части, что позволяет создать быстрый и производительный API для обработки запросов клиентской части.

3. Python для модуля AI-стилизации изображений, что позволяет использовать передовые библиотеки машинного обучения и компьютерного зрения для создания качественных художественных эффектов.

2.2 Выбор среды разработки

Неотъемлемой частью написания программного обеспечения является выбор среды разработки. IDE позволяет не только упрощать работу написания кода, но и позволяет следить за новыми тенденциями в сфере программирования.

Visual Studio Code - легковесный, но мощный редактор кода, который работает на всех основных операционных системах. Он предоставляет встроенную поддержку для JavaScript, TypeScript и Node.js, а также имеет богатую экосистему расширений для поддержки других языков и фреймворков.

Особенности VS Code:

- Интеграция с Git и другими системами контроля версий.

- Отладка кода прямо в редакторе.

- Поддержка IntelliSense для автодополнения кода и предложений.

- Встроенный терминал для выполнения команд.

- Расширяемость через многочисленные плагины и расширения.

2.3 Выбор библиотек и фреймворков

Для разработки проекта были выбраны следующие библиотеки и фреймворки:

Фронтенд:

- React: библиотека JavaScript для создания пользовательских интерфейсов с компонентным подходом.

- TailwindCSS: утилитарный CSS-фреймворк для быстрого создания адаптивных интерфейсов.

- ShadcnUI: набор компонентов пользовательского интерфейса, основанный на Radix UI и стилизованный с помощью TailwindCSS.

- React-Dropzone: библиотека для удобной загрузки файлов методом drag-and-drop.

- TanStack Query: библиотека для управления серверным состоянием и асинхронными запросами.

Бэкенд:

- Express: минималистичный веб-фреймворк для Node.js.

- Drizzle ORM: современная ORM-библиотека для TypeScript с типизацией.

- Sharp: высокопроизводительная библиотека для обработки изображений в Node.js.

AI-стилизация:

- PIL (Python Imaging Library): библиотека для работы с изображениями в Python.

- NumPy: библиотека для научных вычислений с поддержкой многомерных массивов.

- TensorFlow: открытая библиотека машинного обучения для создания нейронных сетей.

3 Описание реализации проекта

3.1 Архитектура приложения

Архитектура приложения редактора изображений с AI-стилизацией построена по принципу клиент-серверного взаимодействия и состоит из трех основных компонентов:

1. Клиентская часть (фронтенд) - отвечает за пользовательский интерфейс, загрузку и предварительную обработку изображений, а также отображение результатов стилизации.

2. Серверная часть (бэкенд) - обрабатывает запросы от клиента, управляет хранением данных и координирует процесс стилизации изображений.

3. Модуль AI-стилизации - реализует алгоритмы преобразования изображений с использованием технологий искусственного интеллекта.

Помимо веб-версии, приложение реализовано в виде настольного программного обеспечения для операционной системы Windows. Настольная версия распространяется в формате исполняемого .exe файла и имеет все те же функциональные возможности, что и веб-версия, но не требует подключения к интернету для обработки изображений, так как все вычисления выполняются локально на компьютере пользователя. Для упаковки веб-приложения в настольное ПО был использован инструмент Electron, который позволяет создавать кроссплатформенные настольные приложения на базе веб-технологий.

3.2 Пользовательский интерфейс

Пользовательский интерфейс приложения разработан с учетом принципов UX/UI дизайна и включает следующие основные элементы:

- Главная страница с описанием возможностей приложения и быстрым доступом к основным функциям.

- Страница редактора изображений с интерфейсом загрузки файлов, выбора стилей и предпросмотра результатов.

- Галерея стилей с примерами различных художественных эффектов.

- Раздел "Мои проекты" для сохранения и управления обработанными изображениями.

Интерфейс полностью переведен на русский язык и адаптирован для различных устройств, от мобильных телефонов до настольных компьютеров.

3.3 Модуль стилизации изображений

Для стилизации изображений в проекте реализовано два подхода:

1. Использование библиотеки PIL (Python Imaging Library) для создания базовых эффектов стилизации через комбинацию фильтров, корректировок цвета, яркости, контрастности и других параметров изображения.

2. Применение нейронных сетей (модель Google Magenta на базе TensorFlow) для продвинутой художественной стилизации изображений в духе известных художников.

Модуль стилизации обеспечивает высокое качество обработки изображений с сохранением разрешения до 1024 пикселей и поддерживает различные форматы файлов (JPEG, PNG, GIF, WEBP).

4 Заключение

В результате выполнения проекта было разработано веб-приложение "Редактор изображений с AI-стилизацией", которое предоставляет пользователям удобный инструмент для художественной обработки изображений с использованием технологий искусственного интеллекта.

Основные достижения проекта:

- Разработан интуитивно понятный пользовательский интерфейс на русском языке.

- Реализована функциональность загрузки и обработки изображений с высоким качеством.

- Создан набор художественных стилей, вдохновленных работами известных художников.

- Обеспечена возможность сохранения и управления обработанными изображениями.

- Создана версия приложения в формате исполняемого .exe файла для использования на персональных компьютерах под управлением Windows.

Приложение удовлетворяет всем требованиям технического задания и может быть использовано широкой аудиторией для создания художественно стилизованных изображений без необходимости специальных навыков в области дизайна и обработки фотографий.