**Промежуточный отчет по программному проекту**

**1. Основные планы и этапы проекта**

1.1 Краткое описание проекта:

Название проекта: "Умная галерея"

Цель проекта: Изучение технологий машинного обучения и их грамотное использование на практике через создание приложения, позволяющего автоматизировать управление фотогалереей. Интерфейс приложения имеет вспомогательное значение и используется для демонстрации функционала.

Краткое описание задач:

1. Разработка модели для поиска дубликатов изображений.
2. Реализация модели фильтрации изображений по запросам (например, поиск фотографий с морем).
3. Создание пользовательского интерфейса для взаимодействия с приложением.
4. Тестирование и оптимизация приложения для работы с большими коллекциями изображений.
5. Подготовка документации и инструкции для пользователей.

1.2 Планы и этапы выполнения проекта:

|  |  |  |  |
| --- | --- | --- | --- |
| Этап проекта | Описание работ | Ожидаемые результаты | Сроки выполнения |
| |  | | --- | | Исследование и планирование |  |  | | --- | |  | | |  | | --- | | Анализ требований, изучение существующих решений, подбор инструментов |  |  | | --- | |  | | |  | | --- | | Подробный план выполнения проекта, выбор технологического стека |  |  | | --- | |  | | 04.12.2024 |
| |  | | --- | | Создание прототипа пользовательского интерфейса |  |  | | --- | |  | | |  | | --- | |  |  |  | | --- | |  |   Разработка первых версий интерфейса, определение основных взаимодействий пользователя с системой | |  | | --- | |  |  |  | | --- | |  |   Базовый интерфейс, демонстрирующий ключевые функции | 06.02.2025 |
| Разработка модели поиска дубликатов | Разработка и обучение нейросетевой модели для удаления дубликатов изображений | Модель, способная эффективно обнаруживать дубликаты | TBD |
| Разработка модели поиска по ключевым словам | Разработка и обучение clip-модели для поиска изображений | Модель, позволяющая искать изображения по описаниям | TBD |
| Интеграция моделей в приложение | Связывание серверной и клиентской частей, разработка API | Рабочее приложение с основным функционалом | TBD |

**2. Используемый технологический стек и его обоснование**

2.1 Перечень используемых технологий:

|  |  |  |
| --- | --- | --- |
| Технология/  Инструмент | Описание | Причины выбора |
| Python | |  | | --- | | Основной язык программирования проекта. Поддерживает множество библиотек для машинного обучения, обработки изображений и работы с данными. |  |  | | --- | |  | | |  | | --- | | Широко распространенный язык с большим сообществом, удобен для разработки ML-приложений, поддерживает интеграцию с различными инструментами анализа данных. |  |  | | --- | |  | |
| PyTorch | |  | | --- | | Фреймворк для работы с нейронными сетями, позволяющий строить, обучать и тестировать модели машинного обучения. |  |  | | --- | |  | | |  | | --- | | Гибкость в разработке моделей, большое количество готовых решений. |  |  | | --- | |  | |
| OpenCV / PyTorchVision | |  | | --- | | Библиотека для обработки изображений, включает широкий набор инструментов для компьютерного зрения. |  |  | | --- | |  | | |  | | --- | | Оптимизирована для работы с изображениями, поддерживает множество алгоритмов для анализа и модификации изображений. |  |  | | --- | |  | |
| Flet | |  | | --- | | Фреймворк для разработки пользовательского интерфейса на Python. |  |  | | --- | |  | | |  | | --- | | Простота использования, подходит для быстрого создания кроссплатформенных GUI-приложений. |  |  | | --- | |  | |
| PostgreSQL | |  | | --- | | Реляционная база данных для хранения информации о пользователях и метаданных фотографий. |  |  | | --- | |  | | |  | | --- | | Надежность, масштабируемость, поддержка сложных запросов и индексации. |  |  | | --- | |  | |
| Git, Docker и другие инструменты разработки | |  | | --- | | Инструменты для контроля версии, контейнеризации и автоматизации процессов разработки. |  |  | | --- | |  | | Обеспечивают удобную совместную работу, воспроизводимость окружения и ускоряют развертывание приложения. |

2.2 Обоснование выбранного технологического стека:

* **Гибкость и масштабируемость:** Python и PyTorch упрощают разработку и улучшение моделей, а PostgreSQL эффективно управляет метаданными фотографий.
* **Оптимизация для работы с изображениями:** OpenCV и TorchVision обеспечивают мощные алгоритмы для поиска дубликатов и классификации.
* **Удобство разработки интерфейса:** Flet позволяет быстро создавать кроссплатформенные GUI-приложения без сложных веб-фреймворков.
* **Надежность и удобство развертывания:** PostgreSQL, Git и Docker обеспечивают хранение данных, контроль версий и простоту развертывания.
* **Совместимость и расширяемость:** Используемые технологии легко интегрируются, позволяя добавлять новые функции и масштабировать приложение.
* **Активное сообщество и поддержка:** PyTorch, OpenCV и PostgreSQL имеют развитую документацию и решения распространенных проблем.
* **Производительность и эффективность:** Оптимизированная работа с изображениями и БД позволяет обрабатывать большие коллекции без задержек.
* **Безопасность:** PostgreSQL поддерживает надежное шифрование и аутентификацию, обеспеч**ивая конфиденциальность данных.**

**3. Критерии оценивания проекта**

| **Критерий** | **Описание** |
| --- | --- |
| Функциональность - Процент выполнения функциональных требований | Процент выполненных задач относительно полного списка функционала. |
| Производительность - Время отклика | Среднее время ответа приложения при запросах от пользователей (должно быть ≤ 1 сек). |
| Документация - Полнота документации (%) | Наличие полного руководства пользователя и документации для разработчиков (≥ 90%). |
| Тестирование - Процент успешных тестов | Успешное прохождение тестов из общего количества (≥ 95%). |
| Производительность - Потребление ресурсов | Среднее потребление памяти не должно превышать 200 МБ, а загрузка CPU — 30%. |

**4. Особые пометки**

* **Гибкость в выборе технологий**: Технологии и инструменты, указанные в текущем плане, могут быть изменены в процессе разработки. Это связано с возможными новыми требованиями, ограничениями или появлением более подходящих решений.
* **Риски масштабирования**: Тестирование с большими коллекциями изображений может выявить дополнительные потребности в оптимизации производительности.
* **Адаптация функционала**: В процессе разработки могут быть внесены изменения в функциональные требования для улучшения пользовательского опыта.
* **Ресурсные ограничения**: Выбор инструментов будет зависеть от доступности вычислительных ресурсов и времени, выделенного на проект.