**План выполнения проекта**

**1. Анализ требований и проектирование**

**1.1. Уточнение задач**

* Обсуждение с заказчиком:
  + Какие форматы файлов нужно обрабатывать? (.txt, .pdf, .docx, .html и др.)
  + Какие NLP-задачи требуются? (суммаризация, извлечение ключевых фраз, классификация и т. д.)
  + Какие модели предпочтительны? (локальные, OpenAI API, Hugging Face и др.)
  + Какая БД нужна? (SQLite, PostgreSQL, MongoDB)
  + Нужен ли веб-интерфейс или CLI достаточно?

**1.2. Техническое проектирование**

* Выбор стека технологий (Python + NLP-библиотеки + БД).
* Определение архитектуры (модули для чтения файлов, NLP-обработки, сохранения в БД).
* Планирование асинхронной/пакетной обработки (если данных много).
* Проектирование структуры БД (таблицы для текстов, метаданных, результатов анализа).

**2. Настройка окружения и прототип (1 неделя)**

**2.1. Подготовка среды разработки**

* Установка Python (3.9+) и зависимостей (pip install transformers spacy sqlalchemy pypdf2 python-docx).
* Настройка виртуального окружения (venv или conda).
* Если используются облачные API (OpenAI, Hugging Face Hub) – получение ключей.

**2.2. Минимальный прототип**

* Написать скрипт, который:
  1. Рекурсивно сканирует папку с файлами.
  2. Читает .txt и .pdf (простейший парсинг).
  3. Передает текст в Hugging Face Pipeline для суммаризации.
  4. Сохраняет результат в SQLite.

**3. Основная разработка**

**3.1. Модуль чтения файлов (file\_reader)**

* Поддержка разных форматов:
  + .txt – стандартное чтение.
  + .pdf – PyPDF2 / pdfplumber (извлечение текста с сохранением структуры).
  + .docx – python-docx.
  + .html – BeautifulSoup.
* Обработка ошибок (битые файлы, кодировки).
* Асинхронное чтение (aiofiles для больших объемов).

**3.2. NLP-обработка (nlp\_processor)**

* Выбор модели (в зависимости от задач):
  + **Суммаризация** – facebook/bart-large-cnn, t5-small.
  + **Извлечение ключевых фраз** – spaCy + KeyBERT.
  + **Классификация** – distilbert-base-uncased.
* Кэширование результатов (если обработка прерывается).
* Поддержка пакетной обработки (batch processing).

**3.3. Работа с БД (db\_manager)**

* Создание таблиц:
  + documents (исходный текст, путь к файлу).
  + summaries (сжатый текст, модель, дата обработки).
  + keywords (ключевые фразы, веса).
* Использование ORM (SQLAlchemy).
* Индексация для быстрого поиска.

**4. Тестирование и оптимизация**

**4.1. Тестирование**

* Юнит-тесты (pytest) для каждого модуля.
* Интеграционные тесты (полный цикл: файл → NLP → БД).
* Обработка "грязных" данных (битые PDF, некорректная кодировка).

**4.2. Оптимизация**

* Ускорение NLP-моделей (квантзация, ONNX Runtime).
* Балансировка между скоростью и качеством (например, distilbert вместо bert-large).
* Параллельная обработка (multiprocessing / asyncio).

**5. Документация и развертывание**

**5.1. Документация**

* README.md с примерами использования.
* Документация API
* Dockerfile

**5.2. Развертывание**

* Локальный запуск:
* Облачное развертывание (Заказчик остановился на серверах Selectel).