Дерево проекта

```
memo_new_api/
├── assets/                          # Статические ресурсы проекта (иконки, логотипы и прочее для визула)
├── backend/                         # Backend сервис (FastAPI)
│   ├── __init__.py                  # Инициализация Python пакета
│   ├── agent_llm_svc.py             # Интеграция агентов с LLM сервисом
│   ├── agent.py                     # Основной модуль агентной системы
│   ├── agents/                      # Агентная архитектура

--- ЭТО ТЕСТОВЫЕ АГЕНТЫ, РЕАЛЬНЫЕ БУДУТ ПИСАТЬСЯ ПОД РЕАЛЬНЫЕ ЗАДАЧИ ---
│   │   ├── base_agent.py            # Базовый класс для всех агентов
│   │   ├── calculation_agent.py     # Агент для математических вычислений
│   │   ├── document_agent.py         # Агент для работы с документами
│   │   ├── langgraph_orchestrator.py # LangGraph оркестратор для управления агентами
│   │   ├── mcp_agent.py             # Агент для интеграции с MCP серверами
│   │   ├── memory_agent.py          # Агент для управления памятью системы
│   │   └── web_search_agent.py      # Агент для поиска в интернете
--- ЭТО ТЕСТОВЫЕ АГЕНТЫ, РЕАЛЬНЫЕ БУДУТ ПИСАТЬСЯ ПОД РЕАЛЬНЫЕ ЗАДАЧИ ---

│   ├── capture_remote_audio.py      # Захват удаленного аудио
│   ├── config/                      # Конфигурация backend
│   │   ├── __init__.py              # Экспорт конфигурации
│   │   ├── config.py                # Основной модуль конфигурации
│   │   ├── config.yml               # YAML конфигурация
│   │   └── server.py                # Настройки сервера FastAPI
│   ├── context_prompts.json          # JSON файл с промптами для контекста (тут хранятся пока что промпты заданные юзерами)
│   ├── context_prompts.py            # Модуль для работы с промптами

--- МОДУЛЬ databese НАХОДИТСЯ В ДОРАБОТКЕ ---
│   ├── database/                    # Работа с базами данных
│   │   ├── __init__.py              # Экспорт модулей БД
│   │   ├── init_db.py               # Инициализация базы данных
│   │   ├── mongodb/                  # MongoDB
│   │   │   ├── __init__.py          # Экспорт MongoDB модулей
│   │   │   ├── connection.py        # Подключение к MongoDB
│   │   │   ├── models.py            # Модели данных MongoDB
│   │   │   └── repository.py        # Место расположения диалогов в MongoDB
│   │   ├── postgresql/              # PostgreSQL (pgvector для RAG-системы)
│   │   │   ├── __init__.py          # Экспорт PostgreSQL модулей
│   │   │   ├── connection.py        # Подключение к PostgreSQL
│   │   │   ├── models.py            # Модели данных PostgreSQL
│   │   │   └── repository.py        # Репозиторий для работы с PostgreSQL
--- МОДУЛЬ databese НАХОДИТСЯ В ДОРАБОТКЕ ---

│   ├── Dockerfile                   # Docker конфигурация для backend
│   ├── document_processor.py        # Обработка и анализ документов
│   ├── env.example                  # Пример переменных окружения
│   ├── llm_client.py                # Клиент для взаимодействия с LLM сервисом
│   ├── llm_settings.json            # Настройки LLM моделей в JSON
│   ├── main.py                      # Точка входа backend приложения
│   ├── mcp_client.py                # Клиент для Model Context Protocol
│   ├── memory/                      # Директория для хранения памяти (Временное решение, пока БД не подключена)
│   ├── memory.py                    # Модуль управления памятью системы (Временное решение, пока БД не подключена)
│   ├── online_transcription.py      # Онлайн транскрипция аудио
│   ├── orchestrator/                # Оркестратор для управления задачами
│   │   ├── __init__.py              # Экспорт оркестратора
│   ├── package-lock.json            # Заблокированные версии npm пакетов
│   ├── requirements.txt             # Python зависимости backend
│   ├── settings.json                # JSON файл с настройками
│   ├── system_audio_capture.py      # Захват системного аудио
│   ├── system_audio.py              # Работа с системным аудио

--- ТЕСТОВЫЕ ИНСТРУМЕНТЫ (тулзы) ДЛЯ ДЕМОНСТРАЦИИ АГЕНТНОЙ АРХИТЕКТУРЫ (будут меняться по мере поступления задач от коллег) ---
│   ├── tools/                       # Инструменты для агентов
│   │   ├── __init__.py              # Экспорт всех инструментов
│   │   ├── agent_tools.py           # Инструменты, вызывающие агентов
│   │   ├── calculation_tools.py     # Инструменты для вычислений
│   │   ├── file_tools.py            # Инструменты для работы с файлами
│   │   ├── system_tools.py          # Инструменты системных операций
│   │   └── web_tools.py             # Инструменты для веб-запросов
--- ТЕСТОВЫЕ ИНСТРУМЕНТЫ (тулзы) ДЛЯ ДЕМОНСТРАЦИИ АГЕНТНОЙ АРХИТЕКТУРЫ (будут меняться по мере поступления задач от коллег) ---

│   ├── transcriber.py               # Базовый модуль транскрипции
│   ├── universal_transcriber.py     # Универсальный транскрибатор
│   ├── uploads/                     # Директория для загруженных файлов
│   ├── utils/                       # Вспомогательные утилиты
```

```
│   │   └── encoding_fix.py          # Исправление проблем с кодировкой
│   ├── voice.py                     # Модуль работы с голосом
│   └── whisperx_transcriber.py      # Транскрибатор на базе WhisperX
├── context_prompts.json             # Глобальные промпты для контекста LLM
├── diarize_models/                  # Веса модели (pyannote) для диаризации спикеров
├── docker-compose.yml               # Основной Docker Compose файл
├── env.main.example                 # Основной пример env файла
├── frontend/                        # React фронтенд приложение
│   ├── build/                       # Собранная версия для продакшена
│   ├── Dockerfile                   # Docker конфигурация для фронтенда
│   ├── nginx.conf                   # Конфигурация Nginx для фронтенда
│   ├── node_modules/                # Установленные npm пакеты
│   ├── package-lock.json            # Заблокированные версии пакетов
│   ├── package.json                 # Зависимости и скрипты Node.js
│   ├── public/                      # Дирректория с файлами иконок, гифок и прочей красоты
│   ├── src/                         # Исходный код React приложения
│   │   ├── App.css                  # Стили главного компонента
│   │   ├── App.test.tsx             # Тесты для App компонента
│   │   ├── App.tsx                  # Главный компонент React
│   │   ├── components/              # React компоненты
│   │   │   ├── AgentArchitectureSettings.tsx # Настройки агентной архитектуры
│   │   │   ├── MessageRenderer.tsx  # Рендерер сообщений
│   │   │   ├── SettingsModal.tsx    # Модальное окно настроек
│   │   │   ├── Sidebar.tsx          # Боковая панель
│   │   │   ├── VoiceIndicator.tsx   # Индикатор голосового ввода
│   │   │   └── settings/            # Компоненты настроек
│   │   │       ├── AboutSettings.tsx # Настройки "О программе"
│   │   │       ├── AgentsSettings.tsx # Настройки агентов
│   │   │       ├── GeneralSettings.tsx # Общие настройки
│   │   │       ├── index.ts         # Экспорт компонентов настроек
│   │   │       ├── ModelsSettings.tsx # Настройки моделей
│   │   │       └── TranscriptionSettings.tsx # Настройки транскрипции
│   │   ├── config/                  # Конфигурация приложения
│   │   │   └── api.ts               # Настройки API клиента
│   │   ├── contexts/                # React контексты
│   │   │   ├── AppContext.tsx       # Главный контекст приложения
│   │   │   └── SocketContext.tsx    # Контекст WebSocket соединений
│   │   ├── index.css                # Глобальные стили
│   │   ├── index.tsx                # Точка входа React приложения
│   │   ├── logo.svg                 # SVG логотип
│   │   ├── pages/                   # Страницы приложения
│   │   │   ├── ChatPage.tsx         # Страница чата
│   │   │   ├── DocumentsPage.tsx    # Страница документов
│   │   │   ├── HistoryPage.tsx      # Страница истории
│   │   │   ├── TranscriptionPage.tsx # Страница транскрипции
│   │   │   ├── UnifiedChatPage.tsx  # Унифицированная страница чата
│   │   │   └── VoicePage.tsx       # Страница голосового ввода
│   │   ├── react-app-env.d.ts       # TypeScript определения для React
│   │   ├── reportWebVitals.ts       # Отчет о производительности
│   │   └── setupTests.ts            # Настройка тестов
│   └── tsconfig.json                # Конфигурация TypeScript
├── llm_settings.json                # Глобальные настройки LLM моделей
├── llm-svc/                         # Микросервис для AI моделей
│   ├── app/                         # Основное приложение сервиса
│   │   ├── __init__.py              # Инициализация пакета
│   │   ├── api/                     # API эндпоинты
│   │   │   ├── __init__.py          # Экспорт API модулей
│   │   │   ├── dependencies.py      # Зависимости для API
│   │   │   └── endpoints/           # API эндпоинты
│   │   │       ├── __init__.py      # Экспорт эндпоинтов
│   │   │       ├── chat.py          # Эндпоинт для чата с LLM
│   │   │       ├── diarization.py   # Эндпоинт для диаризации
│   │   │       ├── health.py        # Проверка здоровья сервиса
│   │   │       ├── models.py        # Эндпоинт для управления моделями
│   │   │       ├── transcription.py # Эндпоинт для транскрипции
│   │   │       ├── tts.py           # Эндпоинт для синтеза речи
│   │   │       └── whisperx.py      # Эндпоинт для WhisperX
│   │   ├── core/                    # Основные модули
│   │   │   ├── __init__.py          # Экспорт core модулей
│   │   │   ├── config.py            # Конфигурация сервиса
│   │   │   └── security.py          # Модуль безопасности
│   │   ├── dependencies/            # Обработчики моделей
│   │   │   ├── __init__.py          # Экспорт зависимостей
│   │   │   ├── diarization_handler.py # Обработчик диаризации
│   │   │   ├── silero_handler.py    # Обработчик Silero TTS
│   │   │   ├── vosk_handler.py      # Обработчик Vosk транскрипции
│   │   │   └── whisperx_handler.py  # Обработчик WhisperX
│   │   ├── llm_dependencies.py       # Зависимости для LLM
│   │   ├── main.py                  # Точка входа LLM сервиса
│   │   ├── models/                  # Модели данных
```

```
│   │   │   ├── __init__.py          # Экспорт моделей
│   │   │   └── schemas.py           # Pydantic схемы данных
│   │   ├── services/                # Бизнес-логика сервисов
│   │   │   ├── __init__.py          # Экспорт сервисов
│   │   │   ├── llama_handler.py     # Обработчик Llama моделей
│   │   │   └── nexus_client.py      # Клиент для Nexus сервиса
│   │   ├── utils/                   # Утилиты приложения
│   │   │   └── downloader.py        # Загрузчик моделей
│   │   └── utils.py                 # Дополнительные утилиты
│   ├── config/                      # Конфигурация llm-svc
│   │   └── config.yml               # Основная конфигурация
│   ├── Dockerfile                   # Docker конфигурация llm-svc
│   ├── env.example                  # Пример переменных окружения
│   ├── LICENSE                      # Лицензия проекта
│   ├── models/                      # Директория для моделей
│   │   └── download_model.py        # Скрипт загрузки моделей
│   ├── requirements.txt             # Python зависимости llm-svc
│   └── test/                         # Тесты для llm-svc
│       ├── __init__.py              # Инициализация тестов
│       ├── conftest.py              # Конфигурация pytest
│       ├── test_api.py              # Тесты API
│       └── test_llama_handler.py    # Тесты обработчика Llama
├── memoai.spec                      # Спецификация для PyInstaller
├── memory/                          # Директория памяти системы
│   └── dialog_history_dialog.json   # История диалогов в JSON (Временное решение, пока не подключены БД)
├── model_small/                     # Маленькая Vosk модель для перевода аудио данных (голоса или аудифайла) в текст
├── models/                          # Директория LLM моделей (ЛОКАЛЬНО, в контуре банка этой дирректории не будет)
│   ├── QVikhr-2.5-1.5B-Instruct-SMPO-Q8_0.gguf # Модель QVikhr
│   └── Qwen3-Coder-30B-A3B-Instruct-Q8_0.gguf # Модель Qwen3 Coder
├── nginx.conf                       # Глобальная конфигурация Nginx
├── pyqt6-6.9.1-cp39-abi3-win_amd64.whl # Wheel файл PyQt6 для Windows
├── requirements.txt                 # Python зависимости проекта
├── settings.json                    # Глобальные настройки проекта
├── silero_models/                   # Глобальные модели Silero TTS (для русского и английского языка)
├── switch_to_llm_svc.py             # Скрипт переключения на LLM сервис
├── uploads/                         # Глобальная директория загрузок
├── venv_312/                        # Файлы виртуального окружения
├── whisperx_models/                 # Модели WhisperX
└── README.md                        # Основная документация проекта
```