**Документация по приложению для анализа климатических данных**

**Описание проекта**

Приложение предназначено для автоматизированного сбора, хранения, анализа и визуализации климатических данных по выбранным городам России. Данные собираются с открытого API (OpenWeatherMap), сохраняются в локальный CSV-файл, который выполняет роль базы данных. Пользователь может через веб-интерфейс просматривать динамические графики климатических показателей.

**Компоненты приложения**

**1. Сбор данных**

* Данные по выбранным городам автоматически запрашиваются с API OpenWeatherMap.
* Параметры данных: температура, ощущаемая температура, давление, влажность, скорость ветра, описание погоды и временная метка.
* Сбор происходит по расписанию (например, каждый час), и новые данные добавляются в CSV-файл.

**2. Хранение данных**

* Для хранения данных используется CSV-файл climate\_data.csv.
* CSV-файл содержит строки с данными о погоде в разных городах и временных отметках.
* Структура CSV:  
  | city | temperature | feels\_like | pressure | humidity | wind\_speed | weather\_description | timestamp |
* CSV выступает в роли базы данных, с которой работает веб-приложение.

**3. Веб-интерфейс (Dash + Flask)**

* Приложение запускается локально на Flask-сервере с фронтендом на Dash.
* Интерфейс позволяет:
  + Выбирать город из списка.
  + Выбирать климатический показатель для анализа (температура, влажность, давление и т.д.).
  + Просматривать графики динамики и распределения показателей.
* Визуализация выполнена с помощью Plotly:
  + Линейные графики для отслеживания изменений во времени.
  + Гистограммы и боксплоты для анализа распределения.
  + Диаграммы рассеивания для анализа взаимосвязей.

**4. Анализ и визуализация данных**

* Все графики интерактивные: можно масштабировать, наводить курсор для получения точных значений.
* Данные динамически фильтруются по выбранному городу и показателю.
* Возможность расширения набора графиков и их настройки.

**Основные файлы и модули**

* app.py — основной файл веб-приложения с интерфейсом Dash и логикой визуализации.
* data\_collector.py (или код парсера) — скрипт для сбора данных с API и сохранения их в CSV.
* climate\_data.csv — файл с накопленными климатическими данными.

**Логика работы**

* При запуске приложения данные из CSV загружаются в DataFrame.
* UI компоненты (выпадающие списки) формируются на основе уникальных городов и показателей.
* Callback-функции реагируют на выбор пользователя, фильтруют данные и строят графики с помощью Plotly.
* В CSV данные добавляются периодически, если запущен парсер.

**Работа с данными**

* Формат времени — UTC, колонка timestamp — datetime.
* Показатели представлены в метрической системе (градусы Цельсия, мм рт. ст., м/с).
* Добавлять новые данные следует через парсер, который обращается к API, получает данные и дозаписывает CSV.

**Планы по дальнейшему развитию**

* Подключение полноценной базы данных MySQL для хранения и управления данными.
* Расширение веб-интерфейса: интерактивные дашборды с несколькими графиками.
* Добавление функции экспорта графиков и отчетов в PDF и Excel.
* Реализация интерфейса для редактирования и удаления данных.
* Улучшение дизайна и UX веб-приложения.