**1. Введение**

**Цель проекта:**  
Создать кроссплатформенное приложение для просмотра прогноза погоды с графиками и таблицами данных.

**Основные функции:**

* Поиск города через API.
* Отображение прогноза погоды на 7 дней.
* Визуализация данных (графики температуры и влажности).
* Вывод дополнительной информации (осадки, качество воздуха).

**Целевая аудитория:**

* Пользователи, которым нужен удобный инструмент для планирования с учётом погодных условий.

**2. Архитектура системы**

Система состоит из двух основных компонентов:

1. **Backend** (логика работы с данными):
   * Взаимодействие с API погоды.
   * Обработка и структурирование данных.
   * Генерация графиков.
2. **Frontend** (пользовательский интерфейс):
   * Форма выбора города.
   * Отображение прогноза погоды.
   * Интерактивные элементы (таблицы, графики).

**Стек технологий:**

* **Язык программирования:** Python 3.x.
* **Библиотеки:**
  + requests – HTTP-запросы к API.
  + pygal – генерация SVG-графиков.
  + cairosvg – конвертация SVG в PNG.
  + flet – создание GUI.
* **API:** WeatherAPI (<https://www.weatherapi.com/>).

**3. Описание модулей**

**3.1 Backend (backend.py)**

**Класс WeatherAPI:**

* **Методы:**
  + search\_cities(query) – поиск городов по запросу.
    - Возвращает список городов в формате: [Страна Регион Город].
  + get\_weather\_forecast(city) – получает прогноз погоды на 7 дней.
    - Возвращает JSON-ответ от API.
  + build\_graphs(forecast\_days) – строит графики температуры и влажности.
    - Генерирует SVG-файлы и конвертирует их в PNG.
  + extract\_weather\_data(forecast\_days) – извлекает данные из ответа API.
    - Формирует словарь с ключами: avgtemp\_c, condition\_text, condition\_icon, precipitation, humidity, air\_quality\_aqi.

**3.2 Frontend (frontend.py)**

**Класс WeatherApp:**

* Управляет навигацией между экранами (/city\_select, /weather).
* Инициализирует API и сохраняет выбранный город.

**Класс CitySelectForm:**

* Форма для ввода города:
  + Поле ввода с автодополнением.
  + Список подсказок (отображается при вводе от 3 символов).
  + Кнопка подтверждения выбора.

**Класс WeatherView:**

* Экран с прогнозом погоды:
  + **Панель 1:** Текущая температура, влажность, погодные условия.
  + **Панель 2:** Таблица с данными на неделю (прокручивается горизонтально).
  + **Панель 3:** Графики температуры и влажности (прокручиваются).
  + **Панель 4:** Случайные стихи (закомментированы в коде).

**3.3 Дополнительные файлы**

* **secret.py:** Содержит API-ключ для WeatherAPI (api\_key1).
* **init.py:** Точка входа в приложение. Настраивает окно, темы и запускает Flet-приложение.

**4. Интеграция с API**

**WeatherAPI:**

* **Endpoints:**
  + search.json – поиск городов.
  + forecast.json – получение прогноза.
* **Параметры запроса:**
  + key – API-ключ.
  + q – название города.
  + days=7 – прогноз на 7 дней.
  + lang=ru – язык ответа (русский).

**Пример ответа API (прогноз):**

json

Copy

Download

{

"forecast": {

"forecastday": [

{

"date\_epoch": 1641081600,

"day": {

"avgtemp\_c": 10.5,

"avghumidity": 75,

"condition": {"text": "Облачно", "icon": "//cdn.weatherapi.com/weather/64x64/day/116.png"},

"daily\_will\_it\_rain": 0

}

}

]

}

}

**5. Интерфейсы**

**5.1 Графический интерфейс (GUI)**

* **Экран выбора города:**  
  + Поле ввода города.
  + Список подсказок.
  + Кнопка "Подтвердить выбор".
* **Экран прогноза погоды:**  
  + Графики температуры и влажности.
  + Таблица с данными на неделю.
  + Кнопка возврата к выбору города (≡).

**5.2 Взаимодействие между компонентами**

