# Техническое задание

## 1. Введение

Проект представляет собой Telegram-бота для получения данных о погоде с возможностью добавления городов в избранное и сохранения информации в базе данных. Бот использует OpenWeatherMap API для получения данных о погоде и базу данных SQLite для хранения истории запросов и избранных городов.

## 2. Цели проекта

1. Предоставление пользователю актуальной информации о погоде по указанному городу.

2. Сохранение данных о погоде в локальной базе данных.

3. Возможность добавления городов в 'Избранное' для быстрого доступа.

4. Интерактивное взаимодействие с пользователем через Telegram.

## 3. Функциональные требования

### 3.1 Команды и функционал бота

* \*\*/start\*\*  
  Запуск бота и приветственное сообщение.  
  Отображение основного меню с кнопками: 'Посмотреть погоду', 'Добавить в избранное', 'Показать избранное', 'О нас'.
* \*\*Посмотреть погоду\*\*  
  Пользователь вводит название города.  
  Бот получает данные о погоде через OpenWeatherMap API и отправляет сообщение с данными о температуре, влажности и описанием погоды.
* \*\*Добавить в избранное\*\*  
  Пользователь вводит название города для сохранения в избранное.
* \*\*Показать избранное\*\*  
  Бот отображает список сохранённых городов и предоставляет возможность получить их погоду.
* \*\*О нас\*\*  
  Отправка информации о назначении бота.

### 3.2 Обработка сообщений

Бот должен обрабатывать текстовые сообщения пользователя, управляя состояниями ввода. Реализованы сценарии для запроса погоды, добавления городов и выбора из избранного.

### 3.3 Данные о погоде

Бот использует OpenWeatherMap API для получения следующих данных о погоде:  
- Температура (в градусах Цельсия)  
- Влажность (в процентах)  
- Описание погоды (например, 'ясно')

### 3.4 База данных

Используется база данных SQLite с двумя таблицами:  
- `weather` — для хранения информации о погоде: город, температура, влажность, описание и дата.  
- `cities` — для хранения избранных городов.

## 4. Нефункциональные требования

1. Производительность: Обработка запросов пользователя за 2–3 секунды.

2. Надёжность: Корректная обработка ошибок при сбоях в API или БД.

3. Интерактивность: Удобное меню с кнопками для пользователя.

4. Хранение данных: Использование SQLite для локального хранения.

5. Безопасность: Скрытие ключей API и токенов бота.

## 5. Технологический стек

1. Язык программирования: Python 3.9+

2. Библиотеки: python-telegram-bot, sqlite3, requests и д.р.

3. База данных: SQLite.

4. API: OpenWeatherMap (текущая погода).

## 6. Ограничения и допущения

1. Требуется активное подключение к интернету.

2. Ключ API должен быть зарегистрирован и актуален.

## 7. Дополнительные требования

1. Логирование ошибок для упрощения диагностики.

2. Тестирование корректности функций работы с БД и API.

3. Подробная документация и комментарии в коде.

## 8. Пример взаимодействия с ботом

\*\*Пользователь:\*\* /start  
\*\*Бот:\*\* Добро пожаловать! Выберите команду из меню.

\*\*Пользователь:\*\* Посмотреть погоду  
\*\*Бот:\*\* Введите название города.

\*\*Пользователь:\*\* Москва  
\*\*Бот:\*\*  
🏙 Город: Москва  
☀ Температура: 10°C  
☔ Влажность: 50%  
✅ Описание: ясно