Техническое задание

на разработку сервиса-агрегатора текущей погоды "Погода"

Общие положения

1.1. Название проекта: "Погода" (или "Погода в среднем")

1.2. Цель проекта: создание простого, удобного сервиса для получения усредненной информации о текущей погоде по выбранному городу, доступного через веб-сайт и мобильное приложение для Android.

1.3. Заказчик: Лукьянов Е.А.

1.4. Исполнитель: Компания ООО “Студенты”

Требования к функциональности:

2.1 Основные функции:

* Ввод названия города пользователем или автоматическое определение местоположения через геолокацию.
* Получение данных о погоде из нескольких источников (Яндекс.Погода, AccuWeather, ГИС-Метео, OpenWeatherMap).
* Обработка и агрегация данных: расчет среднего значения по параметрам (температура, влажность, ветер и др.).
* Отображение усредненных данных пользователю в понятном виде.
* Хранение настроек пользователя (выбранные источники погоды, местоположение) с помощью cookie.
* Автоматическая установка cookie при первом посещении или при отсутствии cookie, с запросом местоположения при необходимости.
* Возможность редактировать настройки (источники данных, местоположение) в любой момент через интерфейс.

2.2 Дополнительные функции:

* Быстрый доступ к последним сохраненным настройкам без повторного ввода (использование cookie).
* Адаптивный дизайн для корректного отображения на ПК и мобильных устройствах.
* Интеграция с Android-приложением для быстрого доступа к данным.
* Технические требования

3.1 Архитектура системы:

Клиентская часть — веб-интерфейс (HTML/CSS/JavaScript), адаптивный дизайн; мобильное приложение — WebView или нативное приложение на Android (Kotlin/Java).

Серверная часть — REST API на Node.js или Python (Flask/Django).

3.2 Интеграция с погодными API:

* Реализовать модули-запросы к API Яндекс.Погода, AccuWeather, ГИС-Метео и OpenWeatherMap согласно их документации.
* Обеспечить обработку ошибок и таймаутов при запросах к сторонним сервисам.

3.3 Обработка данных:

После получения данных от всех источников — вычислять средние значения по каждому параметру и формировать итоговую информацию для отображения.

3.4 Хранение настроек пользователя:

Использовать cookies для хранения выбранных источников погоды и местоположения пользователя.

3.5 Механизм определения местоположения:

При отсутствии cookie — запрашивать у пользователя разрешение на геолокацию или предлагать ввести город вручную; затем сохранять эти данные в cookie.

3.6 Безопасность:

Использовать HTTPS для всех соединений; обеспечить безопасность cookie (HttpOnly, Secure).

Требования к интерфейсу

4.1 Веб-сайт:

Простая форма для ввода города или автоматического определения местоположения; отображение усредненной погоды; кнопки для редактирования настроек.

4.2 Мобильное приложение:

Минимальный интерфейс с возможностью просмотра текущей погоды и доступа к настройкам.

Этапы разработки и сроки выполнения

|  |  |  |
| --- | --- | --- |
| Этап | Описание | Сроки |
| Анализ требований | Детализация функционала | 1 неделя |
| Проектирование архитектуры | Структура системы, схемы взаимодействия | 1 неделя |
| Разработка backend | API, интеграция с погодными сервисами | 3 недели |
| Разработка frontend | Веб-интерфейс | 2 недели |
| Разработка мобильного приложения | Android-приложение | 3 недели |
| Тестирование и отладка | Проверка работоспособности и безопасности | 2 недели |
| Итоговая сдача проекта | Передача заказчику | По завершении |

Документация и поддержка:

* Предоставить документацию по использованию сервиса.
* Обеспечить поддержку в течение 1 месяца после запуска для устранения возможных ошибок.
* Дополнительные требования
* Все компоненты должны быть совместимы с современными браузерами.
* Код должен быть написан согласно стандартам качества и безопасности.
* В случае необходимости — подготовить инструкции по развертыванию сервиса на сервере заказчика.