**1. Минимальный функционал: чтение RSS и вывод на экран**

**Задачи:**

1.1. **Сделать запрос к RSS-каналу**

* Использовать библиотеку feedparser для загрузки и парсинга RSS.
* Вывести список новостей (заголовок, ссылка, дата публикации) в консоль.

1.2. **Создать простейший веб-сервер**

* Настроить Flask для отображения данных.
* Сделать маршрут /, который покажет заголовки новостей в браузере.

**Что узнаем:**

* Основы работы с HTTP-запросами и парсингом XML.
* Как Flask обрабатывает запросы и генерирует ответы.

**2. Интерфейс: вывод новостей в удобной форме**

**Задачи:**

2.1. **Сделать HTML-шаблон**

* Настроить Jinja2 для отображения новостей в виде списка с гиперссылками.

2.2. **Добавить категории и описание**

* В шаблоне показывать категорию и краткое описание каждой новости.

2.3. **Применить CSS для красоты**

* Использовать Bootstrap или свой CSS для стилизации страницы.

**Что узнаем:**

* Как работает система шаблонов Jinja2.
* Базовый CSS и подключение библиотек стилей.

**3. Управление RSS-каналами**

**Задачи:**

3.1. **Добавить функционал добавления RSS-каналов**

* Сделать форму, где пользователь может ввести URL RSS-канала.
* Сохранять список каналов в JSON-файл.

3.2. **Вывод списка каналов**

* Показать на отдельной странице список всех добавленных RSS-каналов.

3.3. **Удаление каналов**

* Добавить кнопку для удаления RSS-канала.

**Что узнаем:**

* Работа с формами (POST-запросы).
* Чтение и запись данных в JSON-файл.
* Основы CRUD-операций.

**4. Обновление новостей автоматически**

**Задачи:**

4.1. **Запуск фонового обновления**

* Настроить планировщик (schedule, APScheduler или аналог) для обновления новостей каждые 10 минут.

4.2. **Хранение только новых новостей**

* Реализовать механизм проверки guid, чтобы сохранять только новые записи.

**Что узнаем:**

* Как запускать фоновую задачу в Python.
* Организация хранения уникальных данных (в файлах).

**5. Сортировка и поиск новостей**

**Задачи:**

5.1. **Сортировка по дате**

* Добавить возможность сортировать новости (сначала новые/старые).

5.2. **Поиск по заголовку**

* Реализовать форму поиска с фильтрацией по заголовку или категории.

**Что узнаем:**

* Обработка GET-запросов (параметры поиска и сортировки).
* Как работать с фильтрацией данных в Python.

**6. Развёртывание приложения в Docker**

**Задачи:**

6.1. **Создать Dockerfile**

* Установить все зависимости для приложения.
* Добавить копирование кода в образ.

6.2. **Настроить запуск в контейнере**

* Прописать CMD для автоматического старта приложения.
* Проверить приложение в локальном контейнере.

6.3. **Добавить поддержку логов**

* Настроить запись логов обновлений в файл (для отладки и мониторинга).

**Что узнаем:**

* Основы работы с Docker.
* Как упаковать Python-приложение для контейнера.

**7. Дополнительные фичи (по желанию)**

**Задачи:**

7.1. **Фильтрация новостей по категориям**

* Возможность выбрать интересующие категории.

7.2. **Пометка избранных новостей**

* Добавить кнопку "Избранное" и показывать эти новости на отдельной странице.

7.3. **Статистика**

* Показать, сколько новостей было добавлено за день/неделю.

**Что узнаем:**

* Дополнительные возможности Flask (сессии, фильтры).
* Как улучшать и оптимизировать проект.