Итоговый проект

Тема: «Краулинг новостных статей с сайта kp.ru».

Сроки: задание открывается 27.01.25. Дедлайн сдачи — 09.02.25.

Задание: в рамках эксперимента социальная платформа N решила ввести новую функциональность — агрегатор новостей. Однако, чтобы выделиться среди остальных площадок, разработчики платформы решили использовать уникальный подход, который заключается в случайности подбора новостных заголовков. Предполагается, что пользователи, которые заходят почитать новости, в наименьшей степени заинтересованы в их узконаправленности. Благодаря такому нововведению компания планирует сэкономить большое количество средств на разработке и поддержке системы рекомендаций новостной ленты.

Для первичного тестирования достаточно иметь в запасе в районе 10 000 информационных блоков за последние сутки, а в качестве источника хватит любого популярного информационного ресурса — таким был выбран ресурс «Комсомольская правда».

Для извлечения понадобились следующие поля:

Название поля	**Описание**	**Обязательно для заполнения**
`title`	Название книги	Да
`description`	Описание	Да
`article_text`	Ссылка на источник данных	Да
`publication_datetime`	Дата и время публикации	Да
`header_photo_url`	URL обложки статьи	Нет
`header_photo_base64`	Base64 обложки статьи	Нет
`keywords`	Ключевые статьи	Да
`authors`	Автор/авторы статьи	Да
`source_url`	Ссылка на источник данных	Да

Требуется реализовать краулер сайта kp.ru (https://www.kp.ru/online/) на фреймворке Scrapy, который сможет обойти онлайн-ленту новостей и собирать заданное в настройках проекта количество постов из ленты. Все посты должны писаться отдельным пайплайном в NoSQL СУБД MongoDB. Предполагается, что при минимальной конфигурации БД можно будет подключить к сервису FastAPI и получать HTML-страницу с n-м количеством статей (например, указанном в query). Для использования браузера рекомендуем взять библиотеку Scrapy Playwright: https://github.com/scrapy-plugins/scrapy-playwright.

Требований к скорости сбора нет — главное, уважать владельцев ресурса и не создать им зловредной нагрузки. По возможности следуй robots.txt и не превышай указанную там максимально возможную скорость сбора.

Для загрузки изображений предлагаем использовать наш ***PhotoDownloaderPipeline*** из файла `pipeline.py`.

Проверить данные можно через сервис FastAPI, находящийся в файле `fastapi_service_news.py`. Инструкция лежит в файле «`Как запустить сервис FastAPI.pdf`».

Система оценивания: десятибалльная.

Критерии оценивания: реализовать паука на Scrapy с использованием библиотеки Scrapy Playwright, собрать сэмпл данных, описать схему данных через scrapy.item, описать собираемые поля в XPath-выражениях, реализовать Pipeline для записи данных во внешний источник данных (MongoDB), подключить PhotoDownloaderPipeline, настроить приоритизацию нескольких Pipeline для правильной работы.

Формат сдачи: код проекта запушить в личный репозиторий GitHub, приложив ссылку на него в окне ниже. В корне проекта оставь сэмпл данных в формате .csv/.jsonlines, в объёме ±1000 объектов, а также скриншот/HTML-страницу ответа прикреплённого веб-приложения.