СИЛАБУС ОСВІТНЬОГО КОМПОНЕНТА
«КОНВЕЄРИ ДАНИХ ТА ХМАРНА ОРКЕСТРАЦІЯ»

Компонент освітньої програми – *вибірковий* (3 кредити)

Освітньо-професійна програма	Комп'ютерні науки та проєктування програмних систем
Спеціальність	F3 Комп'ютерні науки
Галузь знань	F Інформаційні технології
Рівень вищої освіти	другий (магістерський)
Мова навчання	українська
Профайли викладачів	Косович Ігор Тарасович доктор філософії, асистент кафедри математичного моделювання
Контактний тел.	+380372-58-48-25
E-mail:	https://mathmod.chnu.edu.ua/pro-nas/spivrobitnyky/kosovych-ihor-tarasovych/
Сторінка курсу в Moodle	
Консультації	згідно графіку консультацій

АНОТАЦІЯ ОСВІТНЬОГО КОМПОНЕНТА

Мета навчальної дисципліни: сформувати практичне розуміння архітектури data pipelines та навчити студентів самостійно проєктувати, реалізовувати, перевіряти й оркеструвати процеси збору та обробки даних.

Курс знайомить студентів із повним життєвим циклом data pipeline: від отримання даних із API, вебсайтів і файлів до побудови надійних collectors, зберігання даних у локальних та хмарних сховищах, трансформації, перевірки якості та оркестрації процесів за допомогою Apache Airflow.

Окремий акцент зроблено на практичному Data Collection: аналізі HTTP-трафіку, роботі з Postman і Browser DevTools, пошуку undocumented/internal API, вебскрейпінгу, Scrapy та діагностиці проблем зі збором даних. Студенти не лише запускають окремі скрипти, а вчаться перетворювати їх на відтворювані, надійні та контрольовані data workflows.

Принцип курсу: від простого скрипта до повного pipeline. На кожному наступному занятті студент розширює попереднє рішення новими компонентами.

Формат: 12 занять; поєднання коротких теоретичних блоків, live coding, лабораторних робіт та наскрізного практичного проєкту. Рівень – вступний / середній рівень Data Engineering.

Пререквізити: знання Python (змінні, функції, цикли, робота з файлами); базове розуміння табличних даних, робота з Pandas; досвід роботи з хмарними платформами (GCP/AWS).

Основні технології: Python, requests, Postman, Browser DevTools, Burp Suite, BeautifulSoup, Playwright, Scrapy, PostgreSQL, GCS, BigQuery, Pandas, SQL, Apache Airflow.

Очікувані результати навчання. Після завершення курсу студент зможе:

- пояснювати архітектуру data pipeline та обирати між ETL/ELT і batch/streaming підходами;

- працювати з HTTP та REST API, аналізувати request/response, authentication, pagination і rate limits;
- використовувати Postman, Browser DevTools і Burp Suite для діагностики data collection workflows у дозволених середовищах;
- виявляти undocumented/internal API, які використовує вебзастосунок, та відтворювати дозволені запити в Python;
- створювати web scrapers і collectors за допомогою requests, BeautifulSoup, Playwright та Scrapy;
- проєктувати підтримувану структуру data collector із конфігурацією, logging, retries, validation та error handling;
- реалізовувати incremental collection, checkpoints, deduplication та recovery after failure;
- зберігати дані у CSV, JSON, Parquet, PostgreSQL, Google Cloud Storage та BigQuery;
- виконувати базові трансформації й перевірки Data Quality за допомогою Pandas та SQL;
- проєктувати RAW, STAGING, CLEAN та ANALYTICS layers;
- створювати DAG-и в Apache Airflow, налаштовувати dependencies, scheduling, retries, variables і connections;
- об'єднувати collection, validation, transformation і loading у цілісний оркестрований pipeline.

Конкретні datasets, websites та API можуть змінюватися залежно від доступності й навчального контексту без зміни результатів навчання.

НАВЧАЛЬНИЙ КОНТЕНТ ОСВІТНЬОГО КОМПОНЕНТА

МОДУЛЬ 1. DATA PIPELINE FUNDAMENTALS	
Тема 1	**Вступ до Data Pipelines.** Data Pipeline та його компоненти; ETL vs ELT; Batch vs Streaming; CSV / JSON / Parquet; Data Lake / Data Warehouse; Retries, logging, idempotency, observability. *Практична робота:* розбір архітектур; перший API → Python → JSON/CSV pipeline.
МОДУЛЬ 2. DATA COLLECTION, APIS & WEB SCRAPING	
Тема 2	**HTTP, APIs та Postman.** HTTP request/response; Methods і status codes; Headers, params, body, cookies; Authentication; Pagination та rate limits; REST API. *Практична робота:* requests у Postman; перенесення request у Python; API collector із pagination.
Тема 3	**DevTools, undocumented APIs та Burp Suite.** Network / Fetch / XHR; Endpoint, payload, headers, cookies; Undocumented/internal API; Copy as cURL; Burp Proxy, HTTP History, Repeater. *Практична робота:* Website → DevTools → cURL → Postman → Python; аналіз і відтворення дозволеного API request.
Тема 4	**Web Scraping: BeautifulSoup, Playwright та Scrapy.** HTML, DOM, CSS selectors; Static vs dynamic pages; BeautifulSoup; Playwright; Scrapy architecture; Pagination, sessions, rate limiting. *Практична робота:* HTML scraper; dynamic page через Playwright; перший Scrapy Spider; експорт у JSON/CSV.

Тема 5	**Advanced Data Collection.** Причини падіння collector; DNS, CDN, WAF, origin infrastructure; Historical infrastructure research; Censys / certificate / DNS history – для дозволених задач; Anti-bot системи; CAPTCHA як challenge у collection workflow. *Практична робота:* scraping investigation challenge; пошук причини зміни endpoint/infrastructure; стратегія graceful handling challenge та fallback.
МОДУЛЬ 3. BUILDING DATA COLLECTORS	
Тема 6	**Architecture of Data Collectors.** Script vs collector; Project structure; Separation of concerns; .env та secrets; Logging; Timeouts, retries, exceptions; Validation. *Практична робота:* рефакторинг попереднього collector; config + logging + retry + validation.
Тема 7	**Reliable & Incremental Data Collection.** Full vs incremental load; Checkpoints; Timestamps та unique IDs; Deduplication; UPSERT; Exponential backoff; Recovery after failure. *Практична робота:* incremental collector; resume from checkpoint; deduplication і validation.
МОДУЛЬ 4. DATA STORAGE & TRANSFORMATION	
Тема 8	**Data Storage Fundamentals & PostgreSQL.** CSV / JSON / Parquet; Database vs files; PostgreSQL; Schema, PK, indexes; SELECT, WHERE, JOIN, GROUP BY; INSERT / UPSERT. *Практична робота:* Collector → PostgreSQL; створення таблиці; SQL queries; basic UPSERT.
Тема 9	**Cloud Storage & Data Warehouse.** Object Storage; Data Lake / Data Warehouse; GCS; BigQuery; Datasets, tables, schema; RAW / STAGING / CLEAN / ANALYTICS. *Практична робота:* Collector → Parquet/JSON → GCS → BigQuery; створення cloud dataset і query.
Тема 10	**Data Transformation & Data Quality.** Pandas transformations; SQL transformations; Cleaning та normalization; Missing values / duplicates; Schema/row-count/range checks; Intro: dbt та PySpark. *Практична робота:* RAW → Validate → Transform → CLEAN; Data Quality checks; load clean data to BigQuery.
МОДУЛЬ 5. APACHE AIRFLOW & WORKFLOW ORCHESTRATION	
Тема 11	**Apache Airflow Fundamentals.** Навіщо orchestration; DAG / Task / Operator / TaskFlow; Dependencies; Scheduling / Cron; Retries / task states / logs; Variables та Connections. *Практична робота:* перший DAG; Extract → Validate → Transform → Load; failure + retry + log debugging.
Тема 12	**Advanced Airflow & Full Pipeline.** XCom; Sensors; Branching; Catchup / Backfill; Parameters; Connections / credentials; Idempotent tasks; Airflow best practices. *Практична робота:* повний pipeline: collector → validation → transformation → GCS → BigQuery; оркестрація, запуск і перевірка end-to-end workflow.

ФОРМИ, МЕТОДИ ТА ОСВІТНІ ТЕХНОЛОГІЇ НАВЧАННЯ

Методи навчання:

- короткі концептуальні пояснення перед практикою;
- live coding з поясненням прийнятих технічних рішень;
- hands-on лабораторні роботи на реальних або навчальних API, вебсторінках та datasets;
- debugging sessions: пошук причин помилок і відновлення pipeline після failure;
- наскрізний проєкт, який розширюється протягом курсу;
- code review та обговорення trade-offs між різними технічними підходами.

ФОРМИ Й МЕТОДИ КОНТРОЛЮ ТА ОЦІНЮВАННЯ

-Поточний контроль:

Методи контролю:

- перевірка та захист практичних (лабораторних) робіт;
- оцінювання Data Collection mini-project;
- оцінювання Storage & Transformation assignment;
- code review та обговорення технічних рішень;
- короткі перевірки знань ключових концепцій.

Форми поточного контролю: практичні роботи, міні-проєкт зі збору даних, завдання зі зберігання та трансформації даних, короткі перевірки знань, участь у практичних обговореннях.

Підсумковий контроль – залік (захист фінального end-to-end pipeline).

КРИТЕРІЇ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Критерієм підсумкового оцінювання є досягнення студентом мінімальних порогових рівнів оцінок (балів) за кожним передбаченим результатом навчання навчальної дисципліни.

Підсумкова оцінка з дисципліни виставляється за 100-бальною шкалою як сума балів, отриманих студентом за всі види навчальної діяльності, передбачені силабусом. Розподіл балів наведено в таблиці.

Вид навчальної діяльності	**Бали**	**Що оцінюється**
Практичні роботи протягом курсу	35	Коректність коду, відтворюваність, виконання поставленої задачі.
Data Collection mini-project	20	API/scraping collector, структура, pagination, logging, error handling, validation.
Storage & Transformation assignment	15	Зберігання, SQL/Pandas transformations, data layers, Data Quality.
Фінальний Airflow pipeline	25	End-to-end pipeline, DAG structure, dependencies, retries, cloud loading, reproducibility.
Активність / короткі перевірки знань	5	Розуміння ключових концепцій і участь у практичних обговореннях.
Разом	**100**	

Критерії якості практичних робіт:

- код запускається та відтворює заявлений результат;
- структура рішення є зрозумілою та не містить hardcoded secrets;
- передбачені базові error handling, logging та retry mechanisms там, де це доречно;
- дані перевіряються перед подальшою обробкою;
- pipeline не створює неконтрольованих дублікатів та коректно працює при повторному запуску;
- студент може пояснити архітектуру рішення, джерело даних та ключові технічні рішення.

Фінальний практичний результат:

Наприкінці курсу студент має зібрати end-to-end data pipeline, який включає:

- щонайменше одне реальне або навчальне джерело даних (API / website / file);
- Python collector або Scrapy spider;
- validation та базову Data Quality перевірку;
- збереження raw data;
- трансформацію даних;
- завантаження в PostgreSQL та/або GCS / BigQuery;
- Apache Airflow DAG для scheduling та orchestration;
- logging, retries та можливість повторного запуску без неконтрольованого дублювання.

Референсна архітектура курсу: API / Website → Python / Scrapy Collector → RAW Storage (JSON / Parquet / GCS) → Validation & Transformation → PostgreSQL / BigQuery; оркестрація всіх етапів – Apache Airflow.

ПОЛІТИКА ЩОДО АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ

Дотримання політики щодо академічної доброчесності учасниками освітнього процесу при вивченні освітнього компонента регламентовано чинними документами ЧНУ:

- "Кодекс академічної доброчесності та етики Чернівецького національного університету імені Юрія Федьковича" https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/kodeks-akademichnoi-dobrochesnosti-ta-etyky-chernivetskoho-natsionalnoho-universytetu-imeni-yuriia-fedkovycha/
- "Положення про виявлення та запобігання академічного плагіату у Чернівецькому національному університеті імені Юрія Федьковича" https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polozhennia-pro-vyiavlennia-ta-zapobihannia-akademichnomu-plahiatu-u-chernivetskomu-natsionalnomu-universyteti-imeni-yuriia-fedkovycha/
- "Політика використання штучного інтелекту в Чернівецькому національному університету імені Юрія Федьковича" https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polityka-vykorystannia-shtuchnoho-intelektu-v-chernivetskomu-natsionalnomu-universyteti-imeni-yuriia-fedkovycha-zi-zminamy-zatverdzhenymy-vchenoiu-radoiu-chnu-imeni-yuriia-fedkovycha/

Використання здобувачами освіти інструментів штучного інтелекту (ШІ) під час виконання навчальних завдань:

- *допускається* за умови дотримання принципів академічної доброчесності (здобувач освіти несе відповідальність за зміст, достовірність і результати поданої роботи, а у разі використання ШІ зобов'язаний зазначити факт і спосіб його застосування);
- *не допускається* використання ШІ як заміни самостійного виконання завдань або з метою порушення вимог академічної доброчесності;
- *додаткові обмеження чи вимоги* щодо використання ШІ можуть визначатися викладачем в освітньому процесі для конкретних видів навчальної діяльності відповідно до особливостей дисципліни.

Політика освітнього компонента:

- практичні роботи мають бути виконані самостійно або в межах дозволеного формату командної роботи;
- використання AI-асистентів допускається для пояснення, debugging та генерації окремих фрагментів коду, але студент повинен розуміти й уміти пояснити подане рішення;
- код, отриманий з AI або зовнішніх джерел, має бути перевірений перед запуском і адаптований до конкретної задачі;
- у фінальних роботах оцінюється не лише результат, а й відтворюваність, надійність та аргументація технічних рішень.

ІНФОРМАЦІЙНІ РЕСУРСИ

Програмне забезпечення:

- Python 3.x та середовище розробки / Jupyter Notebook;
- Git та GitHub для зберігання й версіонування коду;
- Postman для роботи з API;
- Browser Developer Tools;
- Burp Suite Community Edition для навчального аналізу HTTP-трафіку;
- Scrapy, BeautifulSoup, Playwright, requests;
- PostgreSQL;
- Google Cloud Storage та BigQuery;
- Apache Airflow.

Офіційна документація (основне джерело технічної довідки):

- Python – https://docs.python.org/3/;
- Requests – https://requests.readthedocs.io/;
- Beautiful Soup – https://www.crummy.com/software/BeautifulSoup/bs4/doc/;
- Playwright for Python – https://playwright.dev/python/;
- Scrapy – https://docs.scrapy.org/;
- Pandas – https://pandas.pydata.org/docs/;
- PostgreSQL – https://www.postgresql.org/docs/;
- Google Cloud Storage – https://cloud.google.com/storage/docs;
- BigQuery – https://cloud.google.com/bigquery/docs;
- Apache Airflow – https://airflow.apache.org/docs/;

- Postman – https://learning.postman.com/docs/;
- Burp Suite – https://portswigger.net/burp/documentation;
- Chrome DevTools – https://developer.chrome.com/docs/devtools.

Покликання на робочу програму освітнього компонента