



## Python + Django + AI | Od podstaw | LearnIT

Numer usługi 2026/03/20/182536/3424656

6 500,00 PLN brutto

6 500,00 PLN netto

37,36 PLN brutto/h

37,36 PLN netto/h

157,50 PLN cena rynkowa ⓘ

LEARN IT SPÓŁKA Z  
OGRANICZONĄ  
ODPOWIEDZIALNOŚĆ  
CIA

★★★★★ 4,7 / 5

15 ocen

📄 Usługa szkoleniowa

📺 zdalna w czasie rzeczywistym

🕒 174:00 h

📅 02.07.2026 do 07.12.2026

## Informacje podstawowe

### Kategoria

Informatyka i telekomunikacja / Programowanie

### Identyfikatory projektów

Kierunek - Rozwój, Zachodniopomorskie Bony Szkoleniowe, Regionalny Fundusz Szkoleniowy II

### Grupa docelowa usługi

Program szkoleniowy „Python + Django + AI” został zaprojektowany z myślą o osobach dorosłych, które:

1. **Planują zmianę ścieżki zawodowej** – w szczególności osoby niezadowolone z obecnej sytuacji zawodowej, poszukujące stabilniejszego i lepiej opłacanego zatrudnienia w sektorze IT
2. **Nie posiadają doświadczenia w programowaniu** – kurs jest skierowany do osób początkujących
3. **Są zainteresowane tworzeniem aplikacji internetowych, automatyzacją procesów oraz wykorzystaniem sztucznej inteligencji** – uczestnicy poznają nowoczesne narzędzia wykorzystywane w realnych projektach komercyjnych
4. **Chcą rozwijać się w obszarze backendu, nowoczesnych technologii oraz integracji systemów**
5. **Poszukują elastycznej formy kształcenia dostosowanej do obowiązków zawodowych i rodzinnych**
6. **Są zdeterminowane do zdobycia kompetencji umożliwiających podjęcie pracy na stanowisku Junior Python Developer**
7. **Chcą rozwijać zielone kompetencje**, czyli umiejętność tworzenia energooszczędnych, zrównoważonych i odpowiedzialnych środowisk cyfr

Minimalna liczba uczestników

8

Maksymalna liczba uczestników

30

Data zakończenia rekrutacji

01-07-2026

Forma prowadzenia usługi	zdalna w czasie rzeczywistym
Liczba godzin usługi	174
Podstawa uzyskania wpisu do BUR	Znak Jakości Małopolskich Standardów Usług Edukacyjno-Szkoleniowych (MSUES) - wersja 2.0

# Cel

## Cel edukacyjny

Celem kursu jest kompleksowe przygotowanie uczestników do pracy jako Junior Python Developer, z uwzględnieniem kontekstu zrównoważonego rozwoju (zielone kompetencje).

Uczestnicy uczą się tworzyć aplikacje webowe, pracować w Scrumie, korzystać z frameworków takich jak Django, Flask i FastAPI oraz poznają dobre praktyki programistyczne i sposoby tworzenia energooszczędnych, świadomych rozwiązań IT.

## Efekty uczenia się oraz kryteria weryfikacji ich osiągnięcia i Metody walidacji

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
obsługuje narzędzia Linux i Terminal	<ul style="list-style-type: none"><li>• Tworzy proste skrypty, które zastępują ręczne, powtarzalne czynności, dzięki czemu komputer krócej pracuje na wysokich obrotach → mniej prądu i CO<sub>2</sub>.</li><li>• Wykonuje operacje na plikach partiami i z buforowaniem, więc jest mniej odczytów/zapisów na dysk → mniej energii.</li><li>• Ustawia automatyczne porządki w logach i plikach tymczasowych, by nie magazynować niepotrzebnych danych → mniej miejsca do zasilania i chłodzenia.</li><li>• Potrafi zmierzyć podstawowe zużycie (CPU/RAM/IO) i wskazać, co je obniża → świadome ograniczanie energii.</li></ul>	Test teoretyczny z wynikiem generowanym automatycznie

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
obsługuje system Kontroli Wersji Git	<ul style="list-style-type: none"> <li>• Stosuje listy wykluczeń (.gitignore) i rozsądne przechowywanie plików, aby nie przesyłać i nie trzymać ciężkich, zbędnych danych → mniej transferu i energii.</li> <li>• Łączy drobne zmiany i sprząta stare gałęzie, by skracać czas automatycznych kompilacji → mniej pracy serwerów = mniej CO<sub>2</sub>.</li> <li>• Ustala retencję artefaktów (krócej trzymane paczki/buildy), więc zajmują mniej miejsca → mniej energii na dyski i chłodzenie.</li> <li>• Dobiera prosty sposób wersjonowania, co zmniejsza liczbę niepotrzebnych przebudowań → oszczędność prądu.</li> </ul>	Test teoretyczny z wynikiem generowanym automatycznie
obsługuje podstawy Programowania Python	<ul style="list-style-type: none"> <li>• Dobiera lżejsze struktury danych i unika zbędnych kopii, więc program zużywa mniej pamięci i prądu.</li> <li>• Wybiera rozwiązania o niższej złożoności obliczeniowej (szybsze), co skraca czas pracy komputera → mniej energii.</li> <li>• Czyta/zapisuje pliki rzadziej, w paczkach, więc jest mniej ruchu dyskowego i sieciowego → niższe zużycie energii.</li> <li>• Umie wskazać i poprawić miejsca „marnowania prądu” w kodzie (zbędne pętle, kopiowanie danych).</li> </ul>	Test teoretyczny z wynikiem generowanym automatycznie
obsługuje programowanie obiektowe (OOP) w Pythonie	<ul style="list-style-type: none"> <li>• Projektuje prostsze modele (bez nadmiarowych zależności), dzięki czemu program wykonuje mniej pracy → mniej energii.</li> <li>• Ogranicza „efekty uboczne” w kodzie (enkapsulacja, polimorfizm), więc rzadziej trzeba uruchamiać dodatkowe obliczenia → mniej CO<sub>2</sub>.</li> <li>• Wybiera rozwiązania oszczędne pamięciowo, co zmniejsza obciążenie sprzętu → niższe zużycie prądu.</li> <li>• Umie wyjaśnić, jak decyzje projektowe przekładają się na energię i emisje.</li> </ul>	Test teoretyczny z wynikiem generowanym automatycznie

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
<p>obsługuje prace z wirtualnymi środowiskami</p>	<ul style="list-style-type: none"> <li>• Tworzy środowiska z tylko niezbędnymi bibliotekami (bez „balastu”), więc mniej danych do pobrania i przechowywania → mniej energii.</li> <li>• Aktualizuje i usuwa zależności rozsądnie, co ogranicza ciężkie instalacje i transfer → mniej CO<sub>2</sub>.</li> <li>• Potrafi odtworzyć lekkie środowisko na innym komputerze i porównać zużycie → świadome oszczędzanie energii.</li> <li>• Wyjaśnia, że izolacja środowisk zmniejsza liczbę błędnych, powtarzanych buildów/testów → mniej pracy serwerów.</li> </ul>	<p>Test teoretyczny z wynikiem generowanym automatycznie</p>
<p>obsługuje framework Django</p>	<ul style="list-style-type: none"> <li>• Projektuje widoki i zapytania do bazy tak, aby było ich mniej (np. łączenie zapytań), więc serwer krócej pracuje → mniej prądu.</li> <li>• Stosuje paginację i pamięć podręczną (cache), by nie przetwarzać i nie przysyłać nadmiaru danych → mniej transferu i energii.</li> <li>• Konfiguruje panel administracyjny tak, by nie ładować zbędnych kolumn i rekordów → mniej obliczeń.</li> <li>• Potrafi pokazać metrykami (czas/zużycie) spadek obciążenia po optymalizacji → namacalny efekt środowiskowy.</li> </ul>	<p>Test teoretyczny z wynikiem generowanym automatycznie</p>
<p>obsługuje bazy danych i język SQL</p>	<ul style="list-style-type: none"> <li>• Pisze zapytania z indeksami, limitami i stronicowaniem, dzięki czemu baza nie skanuje całych tabel → mniej energii.</li> <li>• Ustawia klucze/relacje tak, by unikać duplikowania danych → mniej miejsca na dyskach, mniejsze zużycie prądu.</li> <li>• Liczenie i łączenie danych robi po stronie bazy, więc mniej danych wędruje po sieci → mniej energii i CO<sub>2</sub>.</li> <li>• Porównuje plany zapytań i wybiera tańsze energetycznie rozwiązania (krótszy czas, mniej operacji).</li> </ul>	<p>Test teoretyczny z wynikiem generowanym automatycznie</p>

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
<p>obsługuje wzorce projektowe oraz refaktoring i debugowanie kodu</p> <p>obsługuje proces testowania</p>	<ul style="list-style-type: none"> <li>• Wdraża cache/pooling tam, gdzie to skraca pracę systemu → mniej obliczeń i prądu.</li> <li>• Refaktoryzuje kod tak, by wykonywał mniej kroków do tego samego wyniku → mniej energii.</li> <li>• Usuwa „gorące punkty” (miejsca o dużym zużyciu) wykazane w profilu → mierzalny spadek energii.</li> <li>• Ustawia rozsądne logowanie i retencję, aby nie zapisywać nadmiarowych danych → mniej miejsca i prądu.</li> </ul> <ul style="list-style-type: none"> <li>• Uruchamia tylko potrzebne testy (selekcja), więc serwer testowy pracuje krócej → mniej CO<sub>2</sub>.</li> <li>• Stosuje mocki (podmiany), by nie uruchamiać ciężkich usług (baza/sieć) przy każdym teście → mniej energii.</li> <li>• Dbą o współdzielone dane testowe, aby nie powielać dużych zestawów → mniej miejsca i prądu.</li> <li>• Ogranicza generowanie ciężkich raportów/artefaktów, by nie zajmować niepotrzebnie dysków.</li> </ul>	<p>Test teoretyczny z wynikiem generowanym automatycznie</p> <p>Test teoretyczny z wynikiem generowanym automatycznie</p>
<p>obsługuje API: HTTP, REST, WebAPI</p> <p>obsługuje Deploy Aplikacji z wykorzystaniem AWS i Dockera</p>	<ul style="list-style-type: none"> <li>• Projektuje endpointy tak, by wysyłały tylko to, co potrzebne → mniej danych w sieci = mniej energii.</li> <li>• Włącza kompresję, cache i znaczniki ETag, co zmniejsza liczbę odpowiedzi i ich rozmiar → mniej pracy serwerów.</li> <li>• Stosuje poprawne metody/statusy, by uniknąć powtórnych wywołań (retry) → niższe zużycie prądu.</li> <li>• Pokazuje metrykami spadek transferu/czasu odpowiedzi po zmianach → realny efekt ekologiczny.</li> </ul> <ul style="list-style-type: none"> <li>• Buduje lekkie obrazy kontenerów i ustawia limity zasobów, dzięki czemu serwery nie pracują ponad potrzebę → mniej prądu.</li> <li>• Włącza automatyczne skalowanie i właściwy rozmiar maszyn (ani za duże, ani za małe), co ogranicza zbędną pracę serwerów → mniej CO<sub>2</sub>.</li> <li>• Ustawia polityki „lifecycle” dla plików (np. S3), aby nie przechowywać niepotrzebnych danych → mniej energii na dyski.</li> <li>• Prezentuje spadek godzin pracy CPU/transferu po wdrożeniu optymalizacji → mierzalny zysk środowiskowy.</li> </ul>	<p>Test teoretyczny z wynikiem generowanym automatycznie</p> <p>Test teoretyczny z wynikiem generowanym automatycznie</p>

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
obsługuje Flask	<ul style="list-style-type: none"> <li>• Projektuje trasy tak, by było mniej zapytań i mniejsze odpowiedzi → mniej przesyłanych danych = mniej energii.</li> <li>• Stosuje cache szablonów i „streaming” odpowiedzi, aby nie obciążać niepotrzebnie serwera → mniej prądu.</li> <li>• Dodaje tylko konieczne rozszerzenia (biblioteki), by nie „puchła” aplikacja → mniej transferu i miejsca.</li> <li>• Wykazuje spadek zużycia CPU/RAM po optymalizacji → konkretny efekt ekologiczny.</li> </ul>	Test teoretyczny z wynikiem generowanym automatycznie
obsługuje Zadania Rekrutacyjne	<ul style="list-style-type: none"> <li>• Rozwiązuje zadania najkrótszą drogą (niższa złożoność), więc komputer wykonuje mniej pracy → mniej prądu i CO<sub>2</sub>.</li> <li>• Pisze krótki, czytelny kod bez zbędnych działań, co zmniejsza wykorzystanie pamięci i czasu procesora.</li> <li>• Uzasadnia wybór rozwiązania z perspektywy energii („to zużyje mniej zasobów niż alternatywa”).</li> <li>• Pokazuje wyniki pomiarów (czas, pamięć) i wskazuje „zieloną” opcję.</li> </ul>	Test teoretyczny z wynikiem generowanym automatycznie

## Kwalifikacje

### Kompetencje

Usługa prowadzi do nabycia kompetencji.

#### Warunki uznania kompetencji

Pytanie 1. Czy dokument potwierdzający uzyskanie kompetencji lub wyraźnie z nim powiązane inne dokumenty związane ze wsparciem zawierają opis efektów uczenia się?

TAK

Pytanie 2. Czy dokument lub wyraźnie z nim powiązane inne dokumenty związane ze wsparciem potwierdzają, że walidacja została przeprowadzona w oparciu o zdefiniowane w efektach uczenia się kryteria ich weryfikacji i zgodnie z zaplanowanymi metodami walidacji?

TAK

Pytanie 3. Czy dokument lub wyraźnie z nim powiązane inne dokumenty związane ze wsparciem potwierdzają zastosowanie rozwiązań zapewniających rozdzielenie procesów kształcenia i szkolenia od walidacji?

TAK

# Program

## Moduł 1: Podstawy programowania i Pythona

### Lekcja 1

- Rozwój architektury komputerowej – aspekt środowiskowy (zielone kompetencje): wybór i projektowanie rozwiązań o niższym poborze mocy; zrozumienie, że wydajniejsze architektury CPU/GPU zmniejszają zużycie energii i ślad węglowy centrów danych.
- O programowaniu w ogóle – aspekt środowiskowy: promowanie prostoty i unikania overengineeringu, co ogranicza potrzebę nadmiernych zasobów obliczeniowych i zużycia energii.
- Powszechne języki programowania (klasyfikacja, rodzaje) – aspekt środowiskowy: świadomy dobór narzędzi pod kątem efektywności energetycznej i śladu środowiskowego.
- Algorytmy – aspekt środowiskowy: projektowanie algorytmów o mniejszej złożoności obliczeniowej, co bezpośrednio redukuje zużycie energii.
- Reprezentacja danych w komputerze – aspekt środowiskowy: oszczędne gospodarowanie pamięcią (mniej kopii, kompaktowe struktury) zmniejsza zużycie energii.
  - Podstawowe pojęcia algebry logiki – aspekt środowiskowy: logiczna redukcja liczby operacji obniża zapotrzebowanie na energię.

### Lekcja 2: Wprowadzenie do ekosystemu Python

- Pierwsze spotkanie z Pythonem – aspekt środowiskowy: świadome użycie narzędzi o niskim narzucie zasobowym
- Podstawowe cechy języka – aspekt środowiskowy: czytelność i prostota kodu ograniczają koszty utrzymania i nadmierowe uruchomienia.
- Narzędzia programistyczne – aspekt środowiskowy: instalacje „slim”, unikanie zbędnych zależności
- Virtualenv – aspekt środowiskowy: izolacja zależności zapobiega marnotrawstwu zasobów i nadmiernym instalacjom

### Lekcja 3: Podstawy Pythona – część 1

- PEP – aspekt środowiskowy: standardy stylu redukują dług technologiczny i niepotrzebne przebudowy.
- Zmienne; Typy danych; Operatory; Łańcuchy znaków; Mutowalne i niemutowalne typy danych – aspekt środowiskowy: dobór struktury danych pod niski koszt pamięci/CPU, ograniczenie kopii i konwersji, co obniża zużycie energii

### Lekcja 4: Podstawy Pythona – część 2

- Konwersja typów – aspekt środowiskowy: unikanie kosztownych rzutowań
- Funkcje wejścia i wyjścia – aspekt środowiskowy: łączenie operacji (batching) i buforowanie, by ograniczyć I/O energetycznie kosztowne
- Komentarze; Importy – aspekt środowiskowy: dokumentacja ułatwia utrzymanie, lazy-import ogranicza ładowanie zbędnych modułów

### Lekcja 5: Podstawy Pythona – część 3

- Formy operatora przypisania; Formatowanie łańcuchów; Operatory porównania; Instrukcje warunkowe – aspekt środowiskowy: krótsze ścieżki wykonania ograniczają liczbę cykli CPU
- Pętle
- Złożoność algorytmów (Big O) – aspekt środowiskowy: niższa złożoność = mniejsze zużycie energii

### Lekcja 6: Programowanie strukturalne

- Funkcje i argumenty; Zakresy; Adnotacje – aspekt środowiskowy: unikanie nadmiarowych alokacji i dostępu globalnych
- Dokumentowanie kodu – aspekt środowiskowy: dłuższy cykl życia rozwiązań = mniej marnotrawstwa zasobów

### Lekcja 7: Programowanie funkcyjne

- List comprehensions; map/filter/reduce – aspekt środowiskowy: strumieniowe przetwarzanie danych zamiast buforowania

- Dekoratory; Domknięcia; Funkcje lambda/wyższego rzędu

### **Lekcja 8: Obsługa wyjątków**

- Wyjątki; Try/except; Finally/else; Raise; Asercje – aspekt środowiskowy: fail-fast i wczesna walidacja ograniczają liczbę energochłonnych uruchomień

### **Lekcja 9: Praca z tekstem, serializacja i system plików**

- Kodowania – aspekt środowiskowy: kompaktowe formaty i kompresja (np. gzip) zmniejszają transfer i pobór mocy

- Serializacja/Deserializacja; JSON/CSV – aspekt środowiskowy: formaty i tryby strumieniowe ograniczają użycie pamięci i energii

- Wyrażenia regularne

- Iteratory/Generatory – aspekt środowiskowy: lazy evaluation zmniejsza footprint pamięci

## **Moduł 2: Systemy kontroli wersji i programowanie obiektowe**

### **Lekcja 1: Systemy kontroli wersji**

- Git; Komendy; Git Flow; GitHub/GitLab/Bitbucket – aspekt środowiskowy: polityki .gitignore i retencji artefaktów, unikanie binariów w repo i czyszczenie gałęzi zmniejszają zużycie przestrzeni, transfer i energię

### **Lekcja 2: OOP – część 1**

- Klasy/obiekty; Metody/atributy; Enkapsulacja; Dziedziczenie/abstrakcja; Przesłanianie; MRO – aspekt środowiskowy: prostsze modele domenowe i preferencja kompozycji ograniczają złożoność, co zmniejsza zapotrzebowanie na zasoby

### **Lekcja 3: OOP – część 2**

- Metody instancji/klasy/statyczne; Klasy danych; Polimorfizm; Przeciążanie operatorów; Własne wyjątki – aspekt środowiskowy: czytelna obsługa błędów i właściwe API ograniczają liczbę kosztownych restartów i buildów

## **Moduł 3: Bazy danych i SQL**

### **Lekcja 1:**

- Bazy danych, SZBD

- SQL – proste zapytania; Funkcje i procedury – aspekt środowiskowy: indeksy, limity, paginacja i logika bliżej danych redukują transfer i zużycie energii

### **Lekcja 2:**

- Klucze; Indeksy; Relacje; Podzapytania/JOIN; ORM – aspekt środowiskowy: optymalizacja planów zapytań i rozsądny dobór trybów lazy/eager obniżają obciążenie serwera.

### **Lekcja 3:**

- Raw SQL i SQLAlchemy; Aplikacja konsolowa; SQLAlchemy + Alembic – aspekt środowiskowy: migracje incrementalne i zero-downtime ograniczają I/O i czas pracy infrastruktury

## **Moduł 4: Sieć, internet i web development**

### **Lekcja 1:**

- Zasady działania internetu; Architektura klient-serwer; Model OSI/TCP-IP; Protokół HTTP; REST – projektowanie lekkich i wydajnych API – aspekt środowiskowy: cache/CDN/kompresja oraz minimalizacja liczby i rozmiaru żądań zmniejszają ruch sieciowy i zapotrzebowanie energetyczne.

## **Moduł 5: Frameworki Flask i Django**

### **Lekcja 1: Flask – część 1**

- Ćwiczenie: prosta aplikacja; Flask jako framework; psycopg2; Raw SQL; Jinja2 – aspekt środowiskowy: lekkie frameworki, pooling połączeń i cache szablonów zmniejszają zużycie CPU/IO

### **Lekcja 2: Flask – część 2**

- Aplikacja z SQLAlchemy ORM; Zapytania ORM; Alembic – aspekt środowiskowy: profilowanie zapytań, unikanie N+1 i kontrolowane migracje ograniczają obciążenie energetyczne bazy

### **Lekcja 3: Django 1**

- Konfiguracja; PostgreSQL; manage; Aplikacje – aspekt środowiskowy: właściwe LOGGING/DEBUG i re-use połączeń redukuje koszty środowiska

### **Lekcja 4: Django 2**

- Widoki; Routing; ORM; Migracje; Szablony; Formularze; Paginacja; Ćwiczenia – aspekt środowiskowy: paginacja i cache endpointów ograniczają transfer i CPU

### **Lekcja 5: Django 3**

- Normalizacja; Queryset; Seeder/Faker – aspekt środowiskowy: ograniczenie duplikacji danych i świadome generowanie danych testowych zmniejsza ślad dyskowy

### **Lekcja 6: Django 5**

- Admin: widoki/filtry/pola; Admin Actions; Ćwiczenia – aspekt środowiskowy: paginacja i operacje wsadowe redukuje liczbę zapytań i obciążenie serwera

### **Lekcja 7: Django 6**

- Bulk Operations; Transakcje/ACID/poziomy izolacji; Rejestracja i autoryzacja – aspekt środowiskowy: krótsze transakcje i throttling logowania zmniejszają zużycie zasobów

## **Moduł 6: Django REST i Celery**

### **Lekcja 1**

- DRF; Postman; Parametry ścieżki i zapytań; Serializery; Ciasteczka; Ćwiczenie DRF; Testowanie – aspekt środowiskowy: filtrowanie pól, cache i profilowanie endpointów zmniejszają ruch oraz koszty energetyczne

### **Lekcja 2**

- Middleware; JWT; Biblioteki; Rejestracja/Logowanie – aspekt środowiskowy: krótkie TTL tokenów, ochrona przed brute-force i minimalizacja rozmiaru ładunku ograniczają niepotrzebny ruch i obciążenie

### **Lekcja 3**

- Cache; Ćwiczenia – aspekt środowiskowy: wielopoziomowe cache (klient/edge/serwer) redukuje liczbę odwołań do serwera

### **Lekcja 4**

- Ćwiczenia DRF; Swagger/OpenAPI – aspekt środowiskowy: jasny kontrakt API ogranicza błędy i niepotrzebne wywołania

### **Lekcja 5: Celery**

- Praca w tle; Celery Beat; Harmonogramy; Ćwiczenia – aspekt środowiskowy: wsadowe przetwarzanie, back-off retry i okna poza szczytem zmniejszają pobór mocy

## **Moduł 7: Równoległość, asynchroniczność i nowoczesne API**

### **Lekcja 1**

- Równoległe wykonywanie; Proces i wątek; GIL; Wielowątkowość; Multiprocessing; Współpraca między procesami – aspekt środowiskowy: właściwy dobór modelu współbieżności minimalizuje straty energii i nadmiarowe uruchomienia

### **Lekcja 2: Asynchroniczność**

- Korutyny; Event Loop; Futures; asyncio; async/await; Synchronizacja zadań – aspekt środowiskowy: back-pressure, throttling i łączenie zadań ograniczają marnotrawstwo zasobów

### **Lekcja 3: Aiohttp**

- Zapoznanie z Aiohttp; Ćwiczenia – aspekt środowiskowy: I/O nieblokujące zmniejsza koszty energetyczne na żądanie

### **Lekcja 4: FastAPI**

- Zapoznanie z FastAPI; Ćwiczenia – aspekt środowiskowy: szybka walidacja i ograniczanie payloadu redukują transfer i CPU

#### **Lekcja 5: Not Only REST**

- WebSockets; GraphQL; Ćwiczenia – aspekt środowiskowy: utrzymane połączenia i selektywne pola redukują liczbę żądań i wielkość danych

#### **Moduł 8: DevOps, konteneryzacja i CI/CD**

##### **Lekcja 1: DevOps**

- Rola inżyniera DevOps; Wprowadzenie do AWS; EC2; S3 – aspekt środowiskowy: klasy storage z polityką lifecycle ograniczają zużycie dysku i energii.

##### **Lekcja 2: AWS**

- RDS; ELB; Route53; AWS CLI; Wdrażanie w chmurze – aspekt środowiskowy: autoscaling, right-sizing i regiony z energią OZE redukują ślad węglowy.

##### **Lekcja 3: Systemy rozproszone**

- Konteneryzacja i wirtualizacja; Docker jako technologia; Ćwiczenie: aplikacja w kontenerze – aspekt środowiskowy: obrazy „slim”, limity zasobów (cgroups) i pomiar CPU/RAM/IO ograniczają pobór energii.

##### **Lekcja 4: CI/CD**

- Wprowadzenie do CI/CD; GitHub Actions; Inne narzędzia CD – aspekt środowiskowy: warunkowe uruchamianie, cache buildów i limitowanie macierzy jobów redukują koszty energetyczne pipeline'ów

#### **Moduł 9: Przygotowanie do rynku pracy**

##### **Lekcja 1: Próba rozmowy technicznej**

- Próba rozmowy technicznej; Odpowiedzi i retrospektywa; CV; LinkedIn; List motywacyjny; Wyszukiwanie ofert; Porady Centrum Kariery – aspekt środowiskowy: eksponowanie projektów z mierzalnymi oszczędnościami energii (np. spadek CPU-hours, transferu, rozmiaru artefaktów).

##### **Lekcja 2: Obrona projektu dyplomowego**

- Obrona projektu dyplomowego – aspekt środowiskowy (wymóg BUR): prezentacja decyzji projektowych ograniczających ślad środowiskowy wraz z metrykami (CPU-hours, pamięć, IO, transfer) i uzasadnieniem wpływu na cele zrównoważonego rozwoju

##### **Lekcja 3: Walidacja**

- Walidacja za pomocą testu teoretycznego z wynikiem generowanym

# Harmonogram

Liczba pozycji harmonogramu: 123

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<b>1 z 123</b> Podstawy algorytmizacji i programowania   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	02-07-2026	18:00	19:20	01:20
<b>2 z 123</b> Przerwa	Leszek Bartmiński	02-07-2026	19:20	19:35	00:15
<b>3 z 123</b> Podstawy algorytmizacji i programowania   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	02-07-2026	19:35	21:00	01:25
<b>4 z 123</b> Wprowadzenie do ekosystemu Python   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	06-07-2026	18:00	19:20	01:20
<b>5 z 123</b> Przerwa	Leszek Bartmiński	06-07-2026	19:20	19:35	00:15

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<p><b>6 z 123</b>  Wprowadzenie do ekosystemu Python   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.</p>	Leszek Bartmiński	06-07-2026	19:35	21:00	01:25
<p><b>7 z 123</b> Podstawy Pythona – część 1   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.</p>	Leszek Bartmiński	09-07-2026	18:00	19:20	01:20
<p><b>8 z 123</b> Przerwa</p>	Leszek Bartmiński	09-07-2026	19:20	19:35	00:15
<p><b>9 z 123</b> Podstawy Pythona – część 1   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.</p>	Leszek Bartmiński	09-07-2026	19:35	21:00	01:25
<p><b>10 z 123</b>  Podstawy Pythona – część 2   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.</p>	Leszek Bartmiński	13-07-2026	18:00	19:20	01:20
<p><b>11 z 123</b> Przerwa</p>	Leszek Bartmiński	13-07-2026	19:20	19:35	00:15

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<p><b>12 z 123</b></p> Podstawy Pythona – część 2   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	13-07-2026	19:35	21:00	01:25
<p><b>13 z 123</b></p> Podstawy Pythona – część 3   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	16-07-2026	18:00	19:20	01:20
<p><b>14 z 123</b> Przerwa</p>	Leszek Bartmiński	16-07-2026	19:20	19:35	00:15
<p><b>15 z 123</b></p> Podstawy Pythona – część 3   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	16-07-2026	19:35	21:00	01:25
<p><b>16 z 123</b></p> Programowanie strukturalne   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	20-07-2026	18:00	19:20	01:20

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
17 z 123 Przerwa	Leszek Bartmiński	20-07-2026	19:20	19:35	00:15
18 z 123 Programowanie strukturalne   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	20-07-2026	19:35	21:00	01:25
19 z 123 Programowanie funkcyjne   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	23-07-2026	18:00	19:20	01:20
20 z 123 Przerwa	Leszek Bartmiński	23-07-2026	19:20	19:35	00:15
21 z 123 Programowanie funkcyjne   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	23-07-2026	19:35	21:00	01:25

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<p><b>22 z 123</b> Obsługa wyjątków   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.</p>	Leszek Bartmiński	27-07-2026	18:00	19:20	01:20
<p><b>23 z 123</b> Przerwa</p>	Leszek Bartmiński	27-07-2026	19:20	19:35	00:15
<p><b>24 z 123</b> Obsługa wyjątków   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.</p>	Leszek Bartmiński	27-07-2026	19:35	21:00	01:25
<p><b>25 z 123</b> Praca z tekstem, serializacja i system plików   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.</p>	Leszek Bartmiński	30-07-2026	18:00	19:20	01:20
<p><b>26 z 123</b> Przerwa</p>	Leszek Bartmiński	30-07-2026	19:20	19:35	00:15

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<p><b>27 z 123</b> Praca z tekstem, serializacja i system plików   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.</p>	Leszek Bartmiński	30-07-2026	19:35	21:00	01:25
<p><b>28 z 123</b> Systemy kontroli wersji   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.</p>	Leszek Bartmiński	03-08-2026	18:00	19:20	01:20
<p><b>29 z 123</b> Przerwa</p>	Leszek Bartmiński	03-08-2026	19:20	19:35	00:15
<p><b>30 z 123</b> Systemy kontroli wersji   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.</p>	Leszek Bartmiński	03-08-2026	19:35	20:45	01:10
<p><b>31 z 123</b> Programowanie obiektowe – część 1   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.</p>	Leszek Bartmiński	10-08-2026	18:00	19:20	01:20

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<b>32 z 123</b> Przerwa	Leszek Bartmiński	10-08-2026	19:20	19:35	00:15
<b>33 z 123</b> Programowanie obiektowe – część 1   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	10-08-2026	19:35	21:45	02:10
<b>34 z 123</b> Programowanie obiektowe – część 2   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	13-08-2026	18:00	19:20	01:20
<b>35 z 123</b> Przerwa	Leszek Bartmiński	13-08-2026	19:20	19:35	00:15
<b>36 z 123</b> Programowanie obiektowe – część 2   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	13-08-2026	19:35	21:00	01:25

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<b>37 z 123</b> Bazy danych i SQL – część 1   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	17-08-2026	18:00	19:20	01:20
<b>38 z 123</b> Przerwa	Leszek Bartmiński	17-08-2026	19:20	19:35	00:15
<b>39 z 123</b> Bazy danych i SQL – część 1   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	17-08-2026	19:35	21:00	01:25
<b>40 z 123</b> Bazy danych i SQL – część 2   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	20-08-2026	18:00	19:20	01:20
<b>41 z 123</b> Przerwa	Leszek Bartmiński	20-08-2026	19:20	19:35	00:15
<b>42 z 123</b> Bazy danych i SQL – część 2   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	20-08-2026	19:35	21:00	01:25

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<b>43 z 123</b> Bazy danych i SQL – część 3   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	24-08-2026	18:00	19:20	01:20
<b>44 z 123</b> Przerwa	Leszek Bartmiński	24-08-2026	19:20	19:35	00:15
<b>45 z 123</b> Bazy danych i SQL – część 3   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	24-08-2026	19:35	21:00	01:25
<b>46 z 123</b> Praca z internetem i siecią   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	27-08-2026	18:00	19:20	01:20
<b>47 z 123</b> Przerwa	Leszek Bartmiński	27-08-2026	19:20	19:35	00:15
<b>48 z 123</b> Praca z internetem i siecią   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	27-08-2026	19:35	21:00	01:25

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<p><b>49 z 123</b> Flask – część 1   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.</p>	Leszek Bartmiński	31-08-2026	18:00	19:20	01:20
<p><b>50 z 123</b> Przerwa</p>	Leszek Bartmiński	31-08-2026	19:20	19:35	00:15
<p><b>51 z 123</b> Flask – część 1   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.</p>	Leszek Bartmiński	31-08-2026	19:35	21:00	01:25
<p><b>52 z 123</b> Flask – część 2   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.</p>	Leszek Bartmiński	03-09-2026	18:00	19:20	01:20
<p><b>53 z 123</b> Przerwa</p>	Leszek Bartmiński	03-09-2026	19:20	19:35	00:15
<p><b>54 z 123</b> Flask – część 2   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.</p>	Leszek Bartmiński	03-09-2026	19:35	22:00	02:25

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
55 z 123 Django – część 1   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	07-09-2026	18:00	19:20	01:20
56 z 123 Przerwa	Leszek Bartmiński	07-09-2026	19:20	19:35	00:15
57 z 123 Django – część 1   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	07-09-2026	19:35	21:00	01:25
58 z 123 Django – część 2   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	10-09-2026	18:00	19:20	01:20
59 z 123 Przerwa	Leszek Bartmiński	10-09-2026	19:20	19:35	00:15
60 z 123 Django – część 2   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	10-09-2026	19:35	21:00	01:25

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<b>61 z 123</b> Django – część 3   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	14-09-2026	18:00	19:20	01:20
<b>62 z 123</b> Przerwa	Leszek Bartmiński	14-09-2026	19:20	19:35	00:15
<b>63 z 123</b> Django – część 3   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	14-09-2026	19:35	22:00	02:25
<b>64 z 123</b> Django – część 4   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	17-09-2026	18:00	19:20	01:20
<b>65 z 123</b> Przerwa	Leszek Bartmiński	17-09-2026	19:20	19:35	00:15
<b>66 z 123</b> Django – część 4   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	17-09-2026	19:35	21:00	01:25

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<b>67 z 123</b> Django – część 5   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	21-09-2026	18:00	19:20	01:20
<b>68 z 123</b> Przerwa	Leszek Bartmiński	21-09-2026	19:20	19:35	00:15
<b>69 z 123</b> Django – część 5   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	21-09-2026	19:35	21:00	01:25
<b>70 z 123</b> Django – część 6   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	24-09-2026	18:00	19:20	01:20
<b>71 z 123</b> Przerwa	Leszek Bartmiński	24-09-2026	19:20	19:35	00:15
<b>72 z 123</b> Django – część 6   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	24-09-2026	19:35	22:00	02:25

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
73 z 123 Django REST – część 1   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	28-09-2026	18:00	19:20	01:20
74 z 123 Przerwa	Leszek Bartmiński	28-09-2026	19:20	19:35	00:15
75 z 123 Django REST – część 1   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	28-09-2026	19:35	21:00	01:25
76 z 123 Django REST – część 2   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	01-10-2026	18:00	19:20	01:20
77 z 123 Przerwa	Leszek Bartmiński	01-10-2026	19:20	19:35	00:15
78 z 123 Django REST – część 2   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	01-10-2026	19:35	21:00	01:25

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
79 z 123 Django REST – część 3   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	05-10-2026	18:00	19:20	01:20
80 z 123 Przerwa	Leszek Bartmiński	05-10-2026	19:20	19:35	00:15
81 z 123 Django REST – część 3   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	05-10-2026	19:35	21:00	01:25
82 z 123 Django REST – część 4   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	08-10-2026	18:00	19:20	01:20
83 z 123 Przerwa	Leszek Bartmiński	08-10-2026	19:20	19:35	00:15
84 z 123 Django REST – część 4   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	08-10-2026	19:35	22:00	02:25

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<b>85 z 123</b> Celery   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	12-10-2026	18:00	19:20	01:20
<b>86 z 123</b> Przerwa	Leszek Bartmiński	12-10-2026	19:20	19:35	00:15
<b>87 z 123</b> Celery   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	12-10-2026	19:35	21:00	01:25
<b>88 z 123</b> Równoległość – część 1   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	19-10-2026	18:00	19:20	01:20
<b>89 z 123</b> Przerwa	Leszek Bartmiński	19-10-2026	19:20	19:35	00:15
<b>90 z 123</b> Równoległość – część 1   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	19-10-2026	19:35	21:00	01:25

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<p><b>91 z 123</b> Asynchroniczność   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.</p>	Leszek Bartmiński	22-10-2026	18:00	19:20	01:20
<p><b>92 z 123</b> Przerwa</p>	Leszek Bartmiński	22-10-2026	19:20	19:35	00:15
<p><b>93 z 123</b> Asynchroniczność   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.</p>	Leszek Bartmiński	22-10-2026	19:35	21:00	01:25
<p><b>94 z 123</b> Aiohttp – część 1   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.</p>	Leszek Bartmiński	26-10-2026	18:00	19:20	01:20
<p><b>95 z 123</b> Przerwa</p>	Leszek Bartmiński	26-10-2026	19:20	19:35	00:15
<p><b>96 z 123</b> Aiohttp – część 1   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.</p>	Leszek Bartmiński	26-10-2026	19:35	22:00	02:25

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<b>97 z 123</b> FastAPI   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	29-10-2026	18:00	19:20	01:20
<b>98 z 123</b> Przerwa	Leszek Bartmiński	29-10-2026	19:20	19:35	00:15
<b>99 z 123</b> FastAPI   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	29-10-2026	19:35	21:00	01:25
<b>100 z 123</b> Not Only REST   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	02-11-2026	18:00	19:20	01:20
<b>101 z 123</b> Przerwa	Leszek Bartmiński	02-11-2026	19:20	19:35	00:15
<b>102 z 123</b> Not Only REST   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	02-11-2026	19:35	21:00	01:25

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<b>103 z 123</b> DevOps   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	05-11-2026	18:00	19:20	01:20
<b>104 z 123</b> Przerwa	Leszek Bartmiński	05-11-2026	19:20	19:35	00:15
<b>105 z 123</b> DevOps   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	05-11-2026	19:35	21:00	01:25
<b>106 z 123</b> Amazon – podstawowe usługi   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	09-11-2026	18:00	19:20	01:20
<b>107 z 123</b> Przerwa	Leszek Bartmiński	09-11-2026	19:20	19:35	00:15
<b>108 z 123</b> Amazon – podstawowe usługi   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	09-11-2026	19:35	21:00	01:25

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<b>109 z 123</b> Konteneryzacja   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	12-11-2026	18:00	19:20	01:20
<b>110 z 123</b> Przerwa	Leszek Bartmiński	12-11-2026	19:20	19:35	00:15
<b>111 z 123</b> Konteneryzacja   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	12-11-2026	19:35	21:00	01:25
<b>112 z 123</b> CI/CD   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	16-11-2026	18:00	19:20	01:20
<b>113 z 123</b> Przerwa	Leszek Bartmiński	16-11-2026	19:20	19:35	00:15
<b>114 z 123</b> CI/CD   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	16-11-2026	19:35	21:00	01:25

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
115 z 123 Próba rozmowy technicznej   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	19-11-2026	18:00	19:20	01:20
116 z 123 Przerwa	Leszek Bartmiński	19-11-2026	19:20	19:35	00:15
117 z 123 Próba rozmowy technicznej   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	19-11-2026	19:35	22:00	02:25
118 z 123 Obrona projektu dyplomowego   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	23-11-2026	18:00	19:20	01:20
119 z 123 Przerwa	Leszek Bartmiński	23-11-2026	19:20	19:35	00:15

Przedmiot / temat	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
<b>120 z 123</b> Obrona projektu dyplomowego   Zajęcia teoretyczno-praktyczne. Zajęcia w formie wykładu, rozmowy na żywo, chatu oraz współdzielenie ekranu.	Leszek Bartmiński	23-11-2026	19:35	22:00	02:25
<b>121 z 123</b> Walidacja za pomocą testu teoretycznego z wynikiem generowanym automatycznie	-	07-12-2026	18:00	19:20	01:20
<b>122 z 123</b> Przerwa	-	07-12-2026	19:20	19:35	00:15
<b>123 z 123</b> Walidacja za pomocą testu teoretycznego z wynikiem generowanym automatycznie	-	07-12-2026	19:35	21:00	01:25

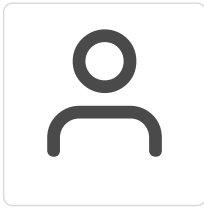
## Cennik

### Cennik

Rodzaj ceny	Cena
<b>Koszt przypadający na 1 uczestnika brutto</b>	6 500,00 PLN
Podmiot uprawniony do zwolnienia z VAT na podstawie art. 43 ust. 1 ustawy o VAT	
<b>Koszt przypadający na 1 uczestnika netto</b>	6 500,00 PLN
<b>Koszt osobogodziny brutto</b>	37,36 PLN
<b>Koszt osobogodziny netto</b>	37,36 PLN

# Prowadzący

Liczba prowadzących: 4



1 z 4

## ANDRZEJ SZESZKO

Jestem profesjonalnym programistą i modelarzem danych z wykształceniem w zakresie danych przestrzennych, UML oraz GIS (tytuł magistra inżyniera).

W ciągu ostatnich kilku lat zdobyłem doświadczenie zawodowe jako programista oraz praktyczną wiedzę z zakresu tworzenia aplikacji webowych z wykorzystaniem różnorodnych narzędzi, ze szczególnym uwzględnieniem języka Python i frameworka Django.

Moje doświadczenie akademickie pozwoliło mi pogłębić znajomość podstaw Pythona oraz rozwinąć umiejętności efektywnego przekazywania wiedzy.

Uczestniczyłem w projektach o znaczeniu krajowym i międzynarodowym, takich jak Geospatial Foundation Theme Governance dla Królestwa Arabii Saudyjskiej oraz Greek Local Spatial Plans Data Model, co umożliwiło mi współpracę z zespołami międzynarodowymi, w których kluczowa była skuteczna komunikacja.

W swojej pracy programistycznej aktywnie stosuję zasady zielonych kompetencji cyfrowych: projektuję rozwiązania o niskim zużyciu zasobów obliczeniowych, optymalizuję zapytania i struktury danych pod kątem efektywności energetycznej, a także wdrażam dobre praktyki zmniejszające ślad węglowy aplikacji – takie jak świadomy dobór algorytmów, minimalizacja zbędnych operacji I/O oraz odpowiedzialne zarządzanie infrastrukturą.

Ma doświadczenie zawodowe zdobyte nie wcześniej niż 5 lat przed datą publikacji usługi w BUR.



2 z 4

## PAWEŁ PIETRASZKO

Ekspert w dziedzinie inżynierii oprogramowania i automatyzacji z wieloletnim doświadczeniem zdobytym zarówno w sektorze komercyjnym, jak i projektach badawczo-rozwojowych. Specjalizuje się w tworzeniu rozwiązań na styku software i hardware, kładąc szczególny nacisk na jakość kodu, wydajność przetwarzania danych oraz niezawodność systemów.

Kluczowe kompetencje techniczne:

Programowanie Full-stack & Systemowe: Biegłość w językach wysokiego poziomu (Python, Java, C#, JavaScript/PHP) oraz systemowych (C++), co pozwala na elastyczne dopasowanie technologii do wymagań projektu.

Quality Assurance & Test Automation: Doświadczenie jako SDET (Software Development Engineer in Test) w projektowaniu zaawansowanych frameworków do testów automatycznych.

Integracja Sprzętowa: Tworzenie aplikacji pracujących blisko fizycznych komponentów i systemów pomiarowych.

Optymalizacja Procesów: Projektowanie i wdrażanie autorskich narzędzi automatyzujących powtarzalne operacje, co realnie przekłada się na oszczędność czasu i redukcję błędów ludzkich.

Podejście zawodowe

W pracy dydaktycznej i projektowej stawia na praktyczne zastosowanie technologii. Dzięki

doświadczeniu w testowaniu, promuje podejście "Quality First", ucząc nie tylko jak pisać kod, ale jak tworzyć systemy odporne na błędy i łatwe w utrzymaniu.

Ma doświadczenie zawodowe zdobyte nie wcześniej niż 5 lat przed datą publikacji usługi w BUR.



3 z 4

## Paweł Wyżykowski

Programista. Technologie: Python, JavaScript, SQL. Programista back-end z ponad 5-letnim doświadczeniem w branży IT, specjalizujący się w komercyjnym tworzeniu aplikacji serwerowych w języku Python.

Posiada praktyczne doświadczenie w projektowaniu, implementacji i utrzymaniu backendów opartych o Python, w tym pracy z frameworkami webowymi, przetwarzaniem danych, integracjami API oraz logiką biznesową aplikacji.

W swojej pracy wykorzystuje nowoczesne podejście do wytwarzania oprogramowania, obejmujące m.in. konteneryzację (Docker), pracę z bazami danych, tworzenie skalowalnych usług backendowych oraz współpracę z zespołami frontendowymi. W codziennej praktyce programistycznej stosuje zasady zielonych kompetencji cyfrowych – projektuje energooszczędny kod, optymalizuje zapytania i struktury danych oraz minimalizuje zbędne operacje obliczeniowe, ograniczając ślad środowiskowy tworzonych aplikacji.

Doświadczenie dydaktyczne oraz praca projektowa pozwalają mu skutecznie przekazywać wiedzę i tłumaczyć zagadnienia programistyczne w sposób praktyczny i zrozumiały dla uczestników szkoleń.

Ma doświadczenie zawodowe zdobyte nie wcześniej niż 5 lat przed datą publikacji usługi w BUR.



4 z 4

## Leszek Bartmiński

Jestem Python Developerem oraz certyfikowanym inżynierem DevOps z ponad 5-letnim doświadczeniem w branży IT. Komercyjnie pracuję niemal wyłącznie w Pythonie, realizując projekty programistyczne i automatyzacyjne. Moje kompetencje obejmują m.in. Docker, Kubernetes, Terraform, Jenkins oraz narzędzia monitorujące (ELK, Grafana), co pozwala mi łączyć wiedzę programistyczną z praktyką inżynierii chmurowej i automatyzacji, w tym projektowania energooszczędnych rozwiązań infrastrukturalnych zgodnych z zasadami zielonych kompetencji cyfrowych.

Pełniłem rolę trenera i mentora – m.in. prowadziłem pierwszą edycję Akademii DevOps w Onwelo SA, w ramach której szkoliłem pracowników i uczestników w zakresie nowoczesnych narzędzi DevOps, programowania w Pythonie i dobrych praktyk w pracy zespołowej. Mam doświadczenie w prowadzeniu zarówno szkoleń technicznych, jak i technicznych-językowych (filologia angielska), co ułatwia mi skuteczne przekazywanie wiedzy, dostosowane do poziomu uczestników.

Jako inżynier informatyki oraz magister filologii angielskiej łączę kompetencje techniczne z umiejętnościami dydaktycznymi i komunikacyjnymi. W pracy trenerskiej stawiam na praktykę, przykłady z realnych projektów oraz aktywizację uczestników, aby zdobyte umiejętności mogli od razu wykorzystać w środowisku zawodowym.

Ma doświadczenie zawodowe zdobyte nie wcześniej niż 5 lat przed datą publikacji usługi w BUR.

# Informacje dodatkowe

## Informacje o materiałach dla uczestników usługi

Uczestnicy kursu otrzymują dostęp do kompletnego zestawu materiałów edukacyjnych, w tym autorskich podręczników, prezentacji, przykładów kodu oraz nagrań wszystkich zajęć, co umożliwia naukę w indywidualnym tempie i powrót do omawianych treści w dowolnym momencie.

Dodatkowo kursanci korzystają ze wskazówek przygotowanych przez Doradcę Kariery, które obejmują m.in. tworzenie skutecznego CV oraz budowanie profesjonalnego profilu na LinkedIn – z uwzględnieniem wymagań branży IT i specyfiki rekrutacji na stanowisko Junior Python Developera.

Nasza usługa została przygotowana w zgodzie z założeniami programu **Zielone Kompetencje**, co oznacza, że w trakcie kursu uczestnicy rozwijają również umiejętności wspierające zrównoważony rozwój, efektywne wykorzystanie zasobów i technologii przyjaznych środowisku – zgodnie z aktualnymi trendami i oczekiwaniami rynku pracy.

### Weryfikacja obecności i frekwencji

Obecność uczestników będzie weryfikowana poprzez:

- system LMS, w którym generowane są listy obecności z każdego spotkania online,
- raporty z platformy (czas logowania, czas aktywności),
- potwierdzenie obecności przez trenera prowadzącego.

Wymagana minimalna frekwencja do zaliczenia kursu wynosi **80%**.

### Dodatkowe elementy monitorowania postępów

- Uczestnicy zobowiązani są do systematycznego wykonywania zadań domowych, które są weryfikowane w systemie LMS.
- Każda sesja jest nagrywana, a nagrania są udostępniane w LMS, co umożliwia weryfikację przebiegu zajęć oraz ewentualne uzupełnienie wiedzy przez uczestników.
- Brak realizacji wymaganych zadań oraz niewystarczająca obecność (poniżej 80%) skutkuje niezaliczeniem szkolenia i brakiem możliwości otrzymania dokumentu potwierdzającego kompetencje.

*Z przyczyn niezależnych od Wykonawcy (np. losowych) harmonogram szkolenia może zostać nieznacznie zmieniony. Wszystkie informacje dostępne w jednostce szkolejcej zostaną przekazane Operatorowi*

Walidacja efektów uczenia się jest wliczona w czas trwania usługi i zostanie przeprowadzona na zakończenie szkolenia w formie testu teoretycznego z wynikiem generowanym automatycznie

### Warunki organizacyjne szkolenia

Szkolenie online w czasie rzeczywistym w małych grupach, z samodzielnym stanowiskiem komputerowym i testem końcowym online.

Nasz kurs to intensywne, praktyczne ścieżka do zawodu Junior Python Developera. Uczymy w czasie rzeczywistym – Godzina szkoleniowa trwa 45 minut, przerwy w usłudze są wliczone w czas usługi rozwojowej. Łącznie 174 godziny dydaktyczne, w tym 20% (35 g) zajęć teoretycznych i 80% (139 g) praktycznych.

### Informacje dodatkowe

Szkolenie realizowane w obszarze technologii informacyjnych (4.2 PRT), obejmujące programowanie w języku Python z zastosowaniem w modelowaniu i symulacji procesów (4.4 PRT) oraz automatyzacji wspierającej transformację cyfrową przedsiębiorstw w kierunku przemysłu 4.0 (4.7 PRT). Usługa powiązana z Programem Rozwoju Technologii Województwa Śląskiego 2019-2030 (PRT) w zakresie: technologii informacyjnych (4.2), modelowania i symulacji procesów (4.4) oraz technologii wspierających przemysł 4.0 (4.7).

**Certyfikat potwierdzający ukończenie szkolenia zostanie wydany uczestnikowi w ciągu 5 dni roboczych od daty zakończenia usługi.**

## Informacje dodatkowe

Zapewniamy:

- ✓ praktyczną wiedzę i umiejętności zgodne z wymaganiami rynku IT
- ✓ wsparcie mentorów i trenerów z doświadczeniem komercyjnym

✔ doradztwo kariery – pomoc w stworzeniu profesjonalnego CV, profilu na LinkedIn i GitHub

✔ zajęcia na żywo online, prowadzone w małych grupach poprzez platformę Zoom.

Szkolenie prowadzone jest przez zespół ekspertów – każdy temat omawiany jest przez dedykowanego trenera, co gwarantuje najwyższą jakość nauki.

Po ukończeniu kursu uczestnik otrzymuje oficjalne zaświadczenie potwierdzające zdobyte kompetencje.

Kurs również dedykowany jest dla osób chcących skorzystać z projektu “Małopolski pociąg do kariery”.

Kolejna edycja usługi przewidziana jest w przeciągu najbliższych **1,5 – 2 miesięcy**.

**Usługa zwolniona z podatku VAT na podstawie art. 43 ust. 1 pkt 29 ustawy z dnia 11 marca 2004 r. o podatku od towarów i usług (Dz.U. 2004 nr 54 poz. 535 z późn. zm.), jeśli jest finansowana w co najmniej 70% ze środków publicznych.**

## Warunki techniczne

Minimalne wymagania sprzętowe obejmują komputer z systemem operacyjnym Windows 10, macOS lub Linux.

Rekomendowana konfiguracja to procesor klasy i5 lub wyższy, co najmniej 8 GB pamięci RAM oraz dysk SSD dla płynnej pracy.

Niezbędne jest również posiadanie kamery internetowej, słuchawek oraz stabilnego łącza internetowego o prędkości min. 3 Mb/s (zarówno dla pobierania, jak i wysyłania danych).

Wszystkie zajęcia – zarówno część teoretyczna, jak i praktyczna (warsztaty i projekty) – realizowane są w formie zdalnej, na żywo, za pośrednictwem platformy Zoom.

## Kontakt



**SIARHEI HLEBKA**

**E-mail** [s.glebko@learnit.com.pl](mailto:s.glebko@learnit.com.pl)

**Telefon** (+48) 571 500 809