



LEARN IT SPÓŁKA Z
OGRANICZONĄ
ODPOWIEDZIALNOŚ
CIĄ

Brak ocen dla tego dostawcy

Python + Django + AI + Zielone Kompetencje | Od podstaw | LearnIT

Numer usługi 2025/07/07/182536/2862118

📍 zdalna w czasie rzeczywistym

🏠 Usługa szkoleniowa

🕒 161 h

📅 28.08.2025 do 26.01.2026

6 500,00 PLN brutto

6 500,00 PLN netto

40,37 PLN brutto/h

40,37 PLN netto/h

Informacje podstawowe

Kategoria

Informatyka i telekomunikacja / Programowanie

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

20

Data zakończenia rekrutacji

27-08-2025

Forma prowadzenia usługi

zdalna w czasie rzeczywistym

Liczba godzin usługi

161

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	• swobodnie porusza się po strukturze systemu plików i wykorzystuje zaawansowane polecenia terminala, • tworzy własne skrypty usprawniające codzienną pracę poprzez automatyzację rutynowych czynności, • potrafi dostosować środowisko pracy do swoich potrzeb lub specyfiki projektu, tak jak robią to profesjonaliści w realnych zespołach IT.	Test teoretyczny z wynikiem generowanym automatycznie
obsługuje system Kontroli Wersji Git	• sprawnie wykonuje operacje na gałęziach projektu, w tym rozgałęzienia i scalania zmian, • potrafi identyfikować i rozwiązywać konflikty wynikające z pracy wielu osób nad tym samym kodem, korzystając z odpowiednich narzędzi, • zna zasady współpracy zespołowej z wykorzystaniem GitHub i aktywnie publikuje oraz zarządza kodem na tej platformie.	Test teoretyczny z wynikiem generowanym automatycznie
obsługuje podstawy Programowania Python	• potrafi wykorzystywać podstawowe struktury danych (listy, krotki, słowniki) w praktycznych zastosowaniach, • zna metody pracy z plikami – umie odczytywać, zapisywać i przetwarzać dane z różnych źródeł, • rozumie i implementuje proste algorytmy służące do rozwiązywania typowych problemów programistycznych.	Test teoretyczny z wynikiem generowanym automatycznie

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
obsługuje programowanie obiektowe (OOP) w Pythonie	• rozumie zasady programowania obiektowego i potrafi tworzyć klasy oraz ich instancje, • wykorzystuje dziedziczenie do organizacji i ponownego użycia kodu, a także stosuje polimorfizm do elastycznego zarządzania zachowaniem obiektów, • stosuje enkapsulację, ukrywając szczegóły implementacji i kontrolując dostęp do danych w celu zachowania spójności i bezpieczeństwa aplikacji.	Test teoretyczny z wynikiem generowanym automatycznie
obsługuje prace z wirtualnymi środowiskami	• potrafi instalować i aktualizować pakiety Python z użyciem narzędzia pip, • tworzy i obsługuje wirtualne środowiska za pomocą takich narzędzi jak virtualenv, dbając o izolację projektów, • skutecznie zarządza zależnościami, kontrolując wersje bibliotek niezbędnych do działania aplikacji, co jest szczególnie istotne przy pracy z wieloma projektami jednocześnie.	Test teoretyczny z wynikiem generowanym automatycznie
obsługuje framework Django	• zna zasady tworzenia aplikacji webowych w Django i potrafi zaprojektować ich podstawową strukturę, • wykorzystuje mechanizm ORM do efektywnej pracy z bazą danych, bez potrzeby pisania zapytań SQL, • umie konfigurować i wykorzystywać Django Admin Panel do zarządzania danymi aplikacji oraz monitorowania jej stanu z poziomu interfejsu administracyjnego.	Test teoretyczny z wynikiem generowanym automatycznie
obsługuje bazy danych i język SQL	• zna podstawy języka SQL i potrafi tworzyć podstawowe zapytania takie jak SELECT, INSERT, UPDATE i DELETE, • stosuje dobre praktyki przy projektowaniu oraz modyfikacji struktur tabel, definiując m.in. klucze główne i obce, • rozumie i wykorzystuje zaawansowane funkcje SQL, w tym operacje JOIN, GROUP BY oraz HAVING do analizy i przetwarzania danych.	Test teoretyczny z wynikiem generowanym automatycznie

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
obsługuje wzorce projektowe oraz refaktoring i debugowanie kodu	• zna i potrafi stosować popularne wzorce projektowe, takie jak Singleton, Factory czy Observer, w praktycznych projektach zespołowych, • pisze przejrzysty, dobrze zorganizowany i łatwy do utrzymania kod zgodnie z zasadami Clean Code, • wykonuje refaktoryzację, usprawniając strukturę istniejącego kodu bez ingerencji w jego działanie.	Test teoretyczny z wynikiem generowanym automatycznie
obsługuje proces testowania	• potrafi tworzyć testy automatyczne z wykorzystaniem PyTest oraz stosować asercje do weryfikacji poprawności działania kodu, • zna techniki mockowania zależności przy użyciu unittest.mock, co pozwala izolować testowany fragment programu, • tworzy testy funkcjonalne, sprawdzające działanie całych procesów w aplikacji, zapewniając kompleksową kontrolę jakości oprogramowania.	Test teoretyczny z wynikiem generowanym automatycznie
obsługuje API: HTTP, REST, WebAPI	• rozumie zasady działania protokołu HTTP, w tym metody żądań (GET, POST, PUT, DELETE) oraz strukturę nagłówków, • potrafi projektować i implementować interfejsy API zgodne z architekturą REST, • zna metody zabezpieczania API i stosuje mechanizmy autoryzacji w celu kontroli dostępu do zasobów aplikacji.	Test teoretyczny z wynikiem generowanym automatycznie
obsługuje Deploy Aplikacji z wykorzystaniem AWS i Dockera	• potrafi tworzyć i zarządzać kontenerami przy użyciu Dockera, rozumiejąc jego kluczowe komponenty i zastosowania, • zna podstawowe usługi platformy AWS i potrafi wdrażać aplikacje w środowisku chmurowym Amazon Web Services, • rozumie ideę Continuous Integration/Deployment i potrafi konfigurować procesy CI/CD z wykorzystaniem narzędzia Jenkins w celu automatyzacji wdrożeń.	Test teoretyczny z wynikiem generowanym automatycznie

Efekty uczenia się	Kryteria weryfikacji	Metoda walidacji
obsługuje Flask	• zna podstawy frameworka Flask, w tym projektowanie tras oraz obsługę żądań i odpowiedzi HTTP, • potrafi korzystać z systemu szablonów Jinja2 do dynamicznego generowania treści HTML, • rozwija aplikacje we Flasku, implementując formularze, dodając nowe funkcjonalności oraz integrując rozszerzenia wspierające rozwój projektu.	Test teoretyczny z wynikiem generowanym automatycznie
obsługuje Zadania Rekrutacyjne	• potrafi rozwiązywać typowe zadania rekrutacyjne z zakresu Pythona i algorytmiki, • analizuje problem, dobiera odpowiednią strukturę danych i pisze czytelny, zoptymalizowany kod, • przygotowuje się do technicznych rozmów kwalifikacyjnych, ćwicząc zadania wymagające logicznego myślenia, debugowania i testowania rozwiązań.	Test teoretyczny z wynikiem generowanym automatycznie

Kwalifikacje

Kompetencje

Usługa prowadzi do nabycia kompetencji.

Warunki uznania kompetencji

Pytanie 1. Czy dokument potwierdzający uzyskanie kompetencji zawiera opis efektów uczenia się?

TAK

Pytanie 2. Czy dokument potwierdza, że walidacja została przeprowadzona w oparciu o zdefiniowane w efektach uczenia się kryteria ich weryfikacji?

TAK

Pytanie 3. Czy dokument potwierdza zastosowanie rozwiązań zapewniających rozdzielenie procesów kształcenia i szkolenia od walidacji?

TAK

Program

Moduł 1: Podstawy programowania i Pythona

Lekcja 1: Podstawy algorytmizacji i programowania

- Rozwój architektury komputerowej
- O programowaniu w ogóle
- Powszechne języki programowania (klasyfikacja, rodzaje)
- Środowisko programistyczne
- Algorytmy
- Systemy liczbowe
- Reprezentacja danych w komputerze (z uwzględnieniem efektywności pamięci i wpływu na zużycie zasobów – zielone kompetencje)
- Podstawowe pojęcia algebry logiki (w kontekście optymalizacji kodu i energooszczędnych rozwiązań)

Lekcja 2: Wprowadzenie do ekosystemu Python

- Pierwsze spotkanie z Pythonem
- Podstawowe cechy języka
- Porównanie Pythona z innymi językami
- Zen Pythona
- Narzędzia programistyczne
- Virtualenv
- Najprostszy program w Pythonie

Lekcja 3: Podstawy Pythona – część 1

- PEP
- Zmienne
- Typy danych
- Operatory
- Łańcuchy znaków
- Mutowalne i niemutowalne typy danych

Lekcja 4: Podstawy Pythona – część 2

- Konwersja typów
- Funkcje wejścia i wyjścia
- Reprezentacja zmiennych w pamięci
- Komentarze
- Importy

Lekcja 5: Podstawy Pythona – część 3

- Formy operatora przypisania
- Formatowanie łańcuchów znaków
- Operatory porównania
- Instrukcje warunkowe
- Pętle
- Złożoność algorytmów (Big O Notation)

Lekcja 6: Programowanie strukturalne

- Funkcje i argumenty
- Zakresy zmiennych
- Adnotacje
- Dokumentowanie kodu

Lekcja 7: Programowanie funkcyjne

- List comprehensions
- Funkcje lambda
- Funkcje wyższego rzędu
- Map, filter, reduce
- Domknięcia i przechwytywanie zmiennych
- Dekoratory

Lekcja 8: Obsługa wyjątków

- Wyjątki
- Try/except

- Bloki finally i else
- Podnoszenie wyjątków (raise)
- Asercje

Lekcja 9: Praca z tekstem, serializacja i system plików

- Kodowania
- Praca z modulem os
- Obsługa plików
- Serializacja i deserializacja
- Praca z danymi zewnętrznymi (JSON, CSV)
- Wyrażenia regularne
- Iteratory i obiekty iterowalne
- Generatory
- Wzorce projektowe

Moduł 2: Systemy kontroli wersji i programowanie obiektowe

Lekcja 1: Systemy kontroli wersji

- Git
- Praca z komendami Git
- Git Flow
- Znajomość serwisów: GitHub, GitLab, Bitbucket

Lekcja 2: Programowanie obiektowe – część 1

- Klasy i obiekty
- Metody, atrybuty, konstruktory
- Enkapsulacja
- Property
- Dziedziczenie i abstrakcja
- Przesłanianie metod i rozszerzenie funkcjonalności
- Wielokrotne dziedziczenie
- MRO

Lekcja 3: Programowanie obiektowe – część 2

- Metody instancji, klasy i statyczne
- Klasy danych
- Polimorfizm
- Przeciążanie operatorów
- Tworzenie własnych wyjątków

Moduł 3: Bazy danych i SQL

Lekcja 1: Bazy danych i SQL – część 1

- Bazy danych
- Systemy zarządzania bazami danych
- Język zapytań SQL, proste zapytania
- Funkcje i procedury

Lekcja 2: Bazy danych i SQL – część 2

- Klucze
- Indeksy
- Relacje
- Podzapytania i JOIN
- ORM

Lekcja 3: Bazy danych i SQL – część 3

- Ćwiczenie praktyczne: użycie raw SQL i SQLAlchemy
- Implementacja aplikacji konsolowej na raw SQL
- Implementacja aplikacji z użyciem SQLAlchemy i Alembic

Moduł 4: Sieć, internet i web development

Lekcja 1: Praca z internetem i siecią

- Zasady działania internetu
- Architektura klient-serwer
- Model OSI i stos protokołów TCP/IP
- Wzorce projektowe (MTV, MVC, MVVM)
- Protokół HTTP
- REST – projektowanie lekkich i wydajnych interfejsów API zgodnych z zasadami zrównoważonego rozwoju

Moduł 5: Frameworki Flask i Django

Lekcja 1: Flask – część 1

- Ćwiczenie: stworzenie prostej aplikacji w Flask
- Użycie Flask jako głównego frameworka
- Praca z psycopg2 do PostgreSQL
- Pisanie surowych zapytań bez ORM
- Użycie Jinja2 do szablonów

Lekcja 2: Flask – część 2

- Ćwiczenie: stworzenie aplikacji w Flask z użyciem SQLAlchemy ORM
- Pisanie zapytań przy użyciu ORM
- System migracji z Alembic – z naciskiem na kontrolę i ograniczenie niepotrzebnych operacji na bazie (efektywność i zielone podejście)

Lekcja 3: Django – część 1

- Django
- Konfiguracja projektu Django
- Podłączenie do PostgreSQL
- Komendy manage
- Aplikacje Django

Lekcja 4: Django – część 2

- Widoki
- Routing i obsługa zapytań
- Modele, Django ORM
- Migracje
- Szablony
- Formularze
- Pagination
- Ćwiczenie: stworzenie prostej aplikacji
- Ćwiczenie: praca z aplikacją Django
- Użycie Django ORM
- Użycie Jinja2

Lekcja 5: Django – część 3

- Normalizacja bazy danych
- Queryset
- Seeder/Faker
- Ćwiczenie: wypełnienie bazy danymi testowymi

Lekcja 6: Django – część 5

- Konfiguracja widoku w panelu administracyjnym
- Sortowanie, filtry i wyszukiwanie
- Inline, wyświetlanie niestandardowych pól
- Admin Actions
- Ćwiczenie: rozbudowa aplikacji

Lekcja 7: Django – część 6

- Operacje zbiorcze (Bulk Operations)
- Transakcje
- ACID
- Poziomy izolacji transakcji
- Ćwiczenie: rozbudowa aplikacji w Django
- Konfiguracja rejestracji i autoryzacji

Moduł 6: Django REST i Celery

Lekcja 1: Django REST – część 1

- Django Rest Framework
- Postman
- Parametry ścieżki i zapytań
- Serializery
- Ciasteczka
- Ćwiczenie: stworzenie aplikacji w DRF
- Testowanie w Django
- Ćwiczenie: stworzenie aplikacji w DRF – projektowanie pod kątem efektywności i minimalizacji zużycia zasobów (zielone kompetencje)

Lekcja 2: Django REST – część 2

- Middleware w Django
- JWT
- Biblioteki
- Rejestracja/Logowanie
- Ćwiczenie: rejestracja i autoryzacja w DRF

Lekcja 3: Django REST – część 4

- Cache
- Ćwiczenie: praca z aplikacją w DRF

Lekcja 4: Django REST – część 5

- Ćwiczenie: praca z aplikacją w DRF
- Dokumentacja API: Swagger, OpenAPI

Lekcja 5: Celery

- Praca w tle z Celery
- Celery Beat
- Wykonywanie zadań wg harmonogramu
- Ćwiczenie: konfiguracja Celery i Celery Beat

Moduł 7: Równoległość, asynchroniczność i nowoczesne API

Lekcja 1: Równoległość – część 1

- Co to jest równoległe wykonywanie
- Proces i wątek
- GIL
- Wielowątkowość
- Multiprocessing
- Współpraca między procesami

Lekcja 2: Asynchroniczność

- Co to jest asynchroniczność
- Korutyny
- Event Loop
- Futures
- Moduł asyncio
- async/await
- Synchronizacja zadań w asyncio – techniki tworzenia responsywnych, zasobooszczędnych aplikacji

Lekcja 3: Aiohttp – część 1

- Zapoznanie z Aiohttp
- Ćwiczenie: stworzenie aplikacji w Aiohttp

Lekcja 4: FastAPI

- Zapoznanie z FastAPI
- Ćwiczenie: prosta aplikacja w FastAPI

Lekcja 5: Not Only REST

- Zapoznanie z WebSockets
- Zapoznanie z GraphQL
- Ćwiczenie: stworzenie aplikacji z użyciem WebSockets

Moduł 8: DevOps, konteneryzacja i CI/CD

Lekcja 1: DevOps

- Rola inżyniera DevOps
- Wprowadzenie do AWS
- EC2
- S3

Lekcja 2: Amazon – podstawowe usługi

- RDS
- ELB
- Route53
- AWS CLI
- Wdrażanie aplikacji w chmurze AWS

Lekcja 3: Systemy rozproszone

- Konteneryzacja i wirtualizacja
- Docker jako technologia – optymalizacja zużycia zasobów, wdrażanie lekkich kontenerów (w duchu zielonych kompetencji)
- Sieci Docker
- docker-compose
- Ćwiczenie: aplikacja w kontenerze – tworzenie rozwiązań wspierających efektywność energetyczną i ograniczenie śladu środowiskowego

Lekcja 4: CI/CD

- Wprowadzenie do CI/CD
- GitHub Actions
- Inne narzędzia CD

Moduł 9: Przygotowanie do rynku pracy

Lekcja 1: Próba rozmowy technicznej

- Przeprowadzenie próbnej rozmowy technicznej
- Odpowiedzi na pytania, retrospektywa
- Tworzenie CV
- Przygotowanie profilu w LinkedIn
- List motywacyjny
- Wyszukiwanie ofert pracy
- Porady Centrum Kariery

Lekcja 3: Obrona projektu dyplomowego

- Obrona projektu dyplomowego

Harmonogram

Liczba przedmiotów/zajęć: 41

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
1 z 41 Podstawy algorytmizacji i programowania	Bartosz Badowski	28-08-2025	18:00	21:00	03:00
2 z 41 Wprowadzenie do ekosystemu Python	Bartosz Badowski	01-09-2025	18:00	21:00	03:00
3 z 41 Podstawy Pythona – część 1	Bartosz Badowski	04-09-2025	18:00	21:00	03:00
4 z 41 Podstawy Pythona – część 2	Bartosz Badowski	08-09-2025	18:00	21:00	03:00
5 z 41 Podstawy Pythona – część 3	Bartosz Badowski	11-09-2025	18:00	21:00	03:00
6 z 41 Programowanie strukturalne	Bartosz Badowski	15-09-2025	18:00	19:00	01:00
7 z 41 Programowanie funkcyjne	Bartosz Badowski	18-09-2025	18:00	21:00	03:00
8 z 41 Obsługa wyjątków	Bartosz Badowski	22-09-2025	18:00	21:00	03:00
9 z 41 Praca z tekstem, serializacja i system plików	Bartosz Badowski	25-09-2025	18:00	21:00	03:00
10 z 41 Systemy kontroli wersji	Bartosz Badowski	29-09-2025	18:00	21:00	03:00
11 z 41 Programowanie obiektowe – część 1	Bartosz Badowski	02-10-2025	18:00	21:00	03:00

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
12 z 41 Programowanie obiektowe – część 2	Bartosz Badowski	06-10-2025	18:00	21:00	03:00
13 z 41 Bazy danych i SQL – część 1	Bartosz Badowski	09-10-2025	18:00	21:00	03:00
14 z 41 Bazy danych i SQL – część 2	Bartosz Badowski	13-10-2025	18:00	21:00	03:00
15 z 41 Bazy danych i SQL – część 3	Bartosz Badowski	16-10-2025	18:00	21:00	03:00
16 z 41 Praca z internetem i siecią	Bartosz Badowski	20-10-2025	18:00	21:00	03:00
17 z 41 Flask – część 1	Bartosz Badowski	23-10-2025	18:00	21:00	03:00
18 z 41 Flask – część 2	Bartosz Badowski	27-10-2025	18:00	21:00	03:00
19 z 41 Django – część 1	Bartosz Badowski	30-10-2025	18:00	21:00	03:00
20 z 41 Django – część 2	Bartosz Badowski	03-11-2025	18:00	21:00	03:00
21 z 41 Django – część 3	Bartosz Badowski	06-11-2025	18:00	21:00	03:00
22 z 41 Django – część 4	Bartosz Badowski	10-11-2025	18:00	21:00	03:00
23 z 41 Django – część 5	Bartosz Badowski	13-11-2025	18:00	21:00	03:00
24 z 41 Django – część 6	Bartosz Badowski	17-11-2025	18:00	21:00	03:00
25 z 41 Django REST – część 1	Bartosz Badowski	20-11-2025	18:00	21:00	03:00
26 z 41 Django REST – część 2	Bartosz Badowski	24-11-2025	18:00	21:00	03:00

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
27 z 41 Django REST – część 3	Bartosz Badowski	27-11-2025	18:00	21:00	03:00
28 z 41 Django REST – część 4	Bartosz Badowski	01-12-2025	18:00	21:00	03:00
29 z 41 Celery	Bartosz Badowski	04-12-2025	18:00	21:00	03:00
30 z 41 Równoległość – część 1	Bartosz Badowski	08-12-2025	18:00	21:00	03:00
31 z 41 Asynchroniczność	Bartosz Badowski	11-12-2025	18:00	21:00	03:00
32 z 41 Aiohttp – część 1	Bartosz Badowski	15-12-2025	18:00	21:00	03:00
33 z 41 FastAPI	Bartosz Badowski	18-12-2025	18:00	21:00	03:00
34 z 41 Not Only REST	Bartosz Badowski	22-12-2025	18:00	21:00	03:00
35 z 41 DevOps	Bartosz Badowski	05-01-2026	18:00	21:00	03:00
36 z 41 Amazon – podstawowe usługi	Bartosz Badowski	08-01-2026	18:00	21:00	03:00
37 z 41 Konteneryzacja	Bartosz Badowski	12-01-2026	18:00	21:00	03:00
38 z 41 CI/CD	Bartosz Badowski	15-01-2026	18:00	21:00	03:00
39 z 41 Próba rozmowy technicznej	Bartosz Badowski	19-01-2026	18:00	21:00	03:00
40 z 41 Obrona projektu dyplomowego	Bartosz Badowski	22-01-2026	18:00	21:00	03:00

Przedmiot / temat zajęć	Prowadzący	Data realizacji zajęć	Godzina rozpoczęcia	Godzina zakończenia	Liczba godzin
41 z 41 Walidacja za pomocą testu z wynikiem generowanym automatycznie	Bartosz Badowski	26-01-2026	18:00	21:00	03:00

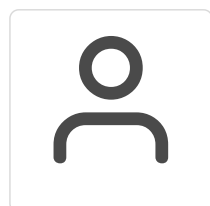
Cennik

Cennik

Rodzaj ceny	Cena
Koszt przypadający na 1 uczestnika brutto	6 500,00 PLN
Koszt przypadający na 1 uczestnika netto	6 500,00 PLN
Koszt osobogodziny brutto	40,37 PLN
Koszt osobogodziny netto	40,37 PLN

Prowadzący

Liczba prowadzących: 1



1 z 1

Bartosz Badowski

Doświadczony Python Developer z ponad 5-letnim stażem w tworzeniu aplikacji webowych i mobilnych, koncentrujący się na backendzie systemów opartych na Pythonie i Django. Świadczone usługi obejmują rozwój i utrzymanie systemów backendowych obsługujących setki tysięcy aktywnych użytkowników. Zakres działań obejmuje tworzenie aplikacji webowych i mobilnych, systemów CMS, paneli administracyjnych oraz integracji z zewnętrznymi usługami, takimi jak systemy płatności, weryfikacja tożsamości czy partnerstwa strategiczne. W ramach realizowanych projektów dostarczane są zarówno MVP, jak i skalowalne, produkcyjne systemy dopasowane do wymagań biznesowych różnych branż. Mam doświadczenie w projektowaniu struktur baz danych, tworzeniu API, integracji z zewnętrznymi usługami (Salesforce, Google APIs, Auth0), systemach rejestracji i uwierzytelniania (OAuth2), testach automatycznych oraz współpracy z zespołami UX/UI, QA i architektami. Pracuję w metodykach Agile (Scrum), regularnie uczestnicząc w ceremoniach zespołowych i planowaniu sprintów. Wcześniej zdobywałem doświadczenie jako Full Stack Developer, realizując MVP w technologii Python/React.

Stack technologiczny: Python, Django, DRF, PostgreSQL, Docker, AWS, Redis, RabbitMQ, Celery, PyTest, Git, CI/CD, Kubernetes, FastAPI, React, TypeScript.

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.

Informacje dodatkowe

Nasz kurs to intensywna, praktyczna ścieżka do zawodu Junior Python Developera. Uczymy w czasie rzeczywistym – uczestnicy pracują nad rzeczywistymi projektami, obserwując na bieżąco sposób działania doświadczonych programistów.

Kurs obejmuje 161 godzin akademickich zajęć.

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 kwalifikacje.

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

Usługa zwolniona z VAT przy finansowaniu minimum 70% ze środków publicznych.

Warunki techniczne

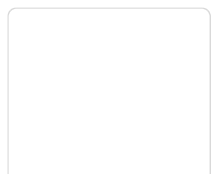
Minimalne wymagania sprzętowe obejmują komputer z systemem operacyjnym Windows 10, macOS lub Linux.

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



Lukasz Gwara

E-mail lukasz.gwara@learnit.com.pl



Telefon (+48) 573 580 990