

| | |
|--|---|
| 13. | Testowanie i integracja aplikacji |
| 14. | Wdrażanie i utrzymanie aplikacji webowych |
| 15. | Zaliczenie wykładu |
| Pracownia specjalistyczna | |
| 1. | Programowanie sieciowe w Pythonie – gniazda (sockets), budowa prostego serwera TCP/HTTP |
| 2. | Programowanie sieciowe w Pythonie – obsługa wielu klientów, parsowanie żądań HTTP, prosty serwer aplikacyjny |
| 3. | Flask – budowa prostej aplikacji webowej (routing, widoki, struktura projektu) |
| 4. | Flask – tworzenie REST API (CRUD, JSON) oraz integracja z bazą danych |
| 5. | Django – utworzenie projektu, modele, migracje i panel administracyjny |
| 6. | Django – widoki, szablony, formularze i obsługa użytkowników |
| 7. | FastAPI – budowa API, routing, dokumentacja OpenAPI |
| 8. | FastAPI – walidacja danych, asynchroniczność i integracja z bazą danych |
| 9. | Wybór tematu projektu |
| 10. | Realizacja projektów |
| 11. | Realizacja projektów |
| 12. | Realizacja projektów |
| 13. | Realizacja projektów |
| 14. | Realizacja projektów |
| 15. | Ocena projektów |
| Metody dydaktyczne (realizacja stacjonarna) | W wykład problemowy; wykład informacyjny Ps programowanie z użyciem komputera |
| Metody dydaktyczne (realizacja zdalna) | - - |
| Forma zaliczenia | W zaliczenie pisemne z pytaniami testowymi i otwartymi Ps ocena wybranych programów realizowanych na zajęciach, ocena zadania projektowego |
| Warunki zaliczenia | Uzyskanie min. 30% z każdego E1-E2, a po spełnieniu tego warunku ostateczna ocena wynika z sumy uzyskanych punktów. Kryteria oceny: W (50 - 60)% punktów - 3.0 [60 - 70)% punktów - 3.5 [70 - 80)% punktów - 4.0 [80 - 90)% punktów - 4.5 [90 - 100)% punktów - 5.0 Uzyskanie min. 30% z każdego E3-E5, a po spełnieniu tego warunku ostateczna ocena wynika z sumy uzyskanych punktów. Kryteria oceny: Ps [0 - 50)% punktów - 2.0 (50 - 60)% punktów - 3.0 (60 - 70)% punktów - 3.5 (70 - 80)% punktów - 4.0 (80 - 90)% punktów - 4.5 (90 - 100)% punktów - 5.0 |

| Symbol efektu | Zakładane efekty uczenia się | Odniesienie do efektów uczenia się zdefiniowanych dla kierunku studiów | | |
|---|--|--|--|-----------------------|
| | | Wiedza | Umiejętności | Kompetencje społeczne |
| Wiedza: student zna i rozumie | | | | |
| E1 | konsepce związane z architekturą aplikacji webowych i zasady budowy aplikacji wykorzystujących bazy danych | DS1_W04 DS1_W08 DS1_W20 DS1_W21 | | |
| E2 | zasady integracji aplikacji z zewnętrznymi usługami oraz podstawy bezpieczeństwa w kontekście aplikacji internetowych | DS1_W04 DS1_W21 | | |
| Umiejętności: student potrafi | | | | |
| E3 | projektować, implementować i wdrażać aplikacje webowe, stosując nowoczesne technologie frontendowe i backendowe oraz dbając o ich wydajność i bezpieczeństwo | | DS1_U02 DS1_U04 DS1_U05 DS1_U12 (H1_U02) DS1_U20 DS1_U21 | |
| E4 | integrować aplikacje z bazami danych oraz wykorzystywać interfejsy API do komunikacji z zewnętrznymi usługami | | DS1_U02 DS1_U07 DS1_U21 | |
| Kompetencje społeczne: student jest gotów do | | | | |
| E5 | przestrzegania zasad bezpieczeństwa i etyki przy projektowaniu aplikacji internetowych oraz odpowiedzialnego podejścia do przetwarzania danych użytkowników | | | DS1_K05 |
| Symbol efektu | Sposób weryfikacji efektu uczenia się | Forma zajęć na której zachodzi weryfikacja | | |
| E1 | zaliczenie pisemne | W | | |
| E2 | zaliczenie pisemne | W | | |
| E3 | rozwiązywanie zadań problemowych | Ps | | |
| E4 | rozwiązywanie zadań problemowych | Ps | | |

| | | |
|---------------------------------|---|-------------------------|
| E5 | rozwiązywanie zadań problemowych | Ps |
| Literatura podstawowa | 1. Charles R. Severance, Python dla Wszystkich. Odkrywanie Danych z Python 3, 2024, online: https://py4e.pl/translations/PL/py4e-pl-print-latest.pdf | |
| | 2. William S. Vincent, Django for Professionals. Production Websites with Python & Django, 2021 | |
| | 3. Mark Lutz, Python. Wprowadzenie, Helion, Gliwice | |
| | 4. David Beazley, Brian K. Jones, Python. Receptury, Helion, Gliwice | |
| | 5. Miguel Grinberg, Flask Web Development. Developing Web Applications with Python, 2018 | |
| Literatura uzupełniająca | 1. Django documentation, online: https://docs.djangoproject.com , (dostęp 2026-05-06) | |
| | 2. Flask documentation, online: https://flask.palletsprojects.com , (dostęp 2026-05-06) | |
| | 3. FastAPI documentation, online: https://fastapi.tiangolo.com , (dostęp 2026-05-06) | |
| | 4. Django REST Framework documentation, online: https://www.django-rest-framework.org/ , (dostęp 2026-05-06) | |
| | 5. Mozilla Foundation, MDN Web Docs - HTTP, online: https://developer.mozilla.org/en-US/docs/Web/HTTP , (dostęp 2026-05-06) | |
| Koordynator przedmiotu: | dr inż. Tomasz Łukaszyk | Data: 06/05/2026 |