

ZSP2.CKZ.ZP.26.59/23.10.2025

Załącznik Nr 8 – 8.1. – 8.3. do SWZ – Część Nr: 1, 2, 3

„Usługi szkolenia specjalistycznego dla nauczycieli” w ramach projektu pn.: „Kompetentny zawodowiec – satysfakcja dla siebie i innych” - IV

OPIS PRZEDMIOTU ZAMÓWIENIA

Część Nr 1, Nr 2, Nr 3

OPIS PRZEDMIOTU ZAMÓWIENIA

Część Nr 1: (26. 2.3.) Kurs „Tworzenie aplikacji desktopowych WPF i .NET”

Skierowany do 2 nauczycieli

Czas trwania: 24h (4dni robocze*6 godzin)

Termin planowany: wg wskazań Zamawiającego II-VI w 2026 r.

Sposób prowadzenia: on-line

Program szkolenia:

1. Teoretyczne omówienie uczestnikom szkolenia problemu w sposób usystematyzowany, analityczny przed przystąpieniem do wykonywania ćwiczenia.(Wprowadzenie do WPF i .Net Core, Architektura aplikacji

WPF, Porównanie WPF z innymi technologiami do tworzenia aplikacji desktopowych, możliwości i ograniczenia WPF.)

2. Kontrolki, style, szablony. (Omówienie: tworzenia, stosowania, wprowadzania stylów i szablonów kontrolki WPF)

3. Zdarzenia myszy i klawiatury

4. Budowa interfejsu użytkownika (XAML – jako język opisu interfejsu – tworzeni interfejsu, omówienie przestrzeni nazw oraz kontrolki w XAML, stylowanie i theming z użyciem XAML)

5. Kontrolki WPF (omówienie kontrolki WPF, praca z nimi, kontrolki do nawigacji i wprowadzania danych)

6. Serwisy sieciowe i REST API – integracja(praca z http Client do komunikacji z API, pobieranie i serializacja danych z API, Wyświetlanie pobranych danych w interfejsie użytkownika, Obsługa błędów i wyjątków podczas wywoływania API.)

7. Korzystanie z ItemTemplate zastosowaniu widoków listy danych (kontrolki do wyświetlania list danych, stworzenie własnego ItemTemplate, implementacja widoków listy danych z wykorzystaniem ItemTemplate, obsługa zdarzeń w kontekście listy danych)

8. Data Binding w WPF (konceptcja data Winding, Two-way, one-way, i one-time binding, konwersje i walidacje w data binding, implementacja data binding w XAML)

9. Integracja z bazami danych WPF (o Podstawy pracy z Entity Framework Core, tworzenie modeli i konfiguracja bazy danych, praca z relacjami między encjami, wykonywanie operacji CRUD na bazie danych)

10. Wprowadzenie do wzorca architektonicznego MVVM (wzorzec MVVM i jego korzyści, implementacja modelu, widoku i ViewModelu, komunikacja między warstwami w MVVM, wykorzystanie ICommand do obsługi zdarzeń)

11. Szkolenie z programowania musi się odbywać w środowisku programistycznym z danego zagadnienia. Na zajęciach uczestnicy programują pod okiem osoby prowadzącej szkolenie. Osoba prowadząca powinna być praktykiem z co najmniej 5-letnim doświadczeniem w przeprowadzaniu kursów i pisanii kodów w czasie rzeczywistym. Wprowadzenie teoretyczne musi

być dostarczone przed lekcją tak aby uczestnicy mogli się z tym zapoznać. Lekcja właściwa musi być ćwiczeniem a trener – programujący w czasie rzeczywistym.

12. Szkolenie powinno odbywać się na poziomie dla nauczycieli szkoły ponadpodstawowej (mając w perspektywie przygotowanie do egzaminu info.04 technika programisty). Materiały np. zadania do wykonania powinny być dostosowane do poziomu technika programisty lub wykraczać poza jego poziom.

13. Wykształcenie prowadzącego dyplom studiów wyższych o kierunkach informatycznych lub pokrewnych np. elektroniczne, elektrotechniczne

3. Obowiązki Wykonawcy:

- Przygotowanie i przekazanie uczestnikom kursu materiałów szkoleniowych w postaci papierowej lub elektronicznej.
- Przekazanie uczestnikom zajęć materiałów teoretycznych wyczerpujących program szkolenia przed rozpoczęciem zajęć, aby uczestnicy szkolenia mogli się z nimi zapoznać
- przygotowanie i przekazanie programu zajęć uwzględniający minimum: nazwę zajęć, łączną liczbę godzin, czas trwania i sposób organizacji zajęć, miejsce realizacji zajęć, imię i nazwisko osoby prowadzącej zajęcia, cele kształcenia, opis efektów uczenia się, konspekt zajęć, sposób i formę zaliczenia
- przygotowanie i przekazanie harmonogramu zajęć uwzględniający minimum: termin i miejsce realizacji zajęć, nazwę zajęć, imię i nazwisko wykładowcy i godziny realizacji zajęć. Harmonogram musi zostać uzgodniony z Zamawiającym niezwłocznie po podpisaniu umowy.
- wydanie uczestnikom szkolenia certyfikatu/zaświadczenia, będącego potwierdzeniem nabycia kompetencji uzyskanych w wyniku formalnej oceny. Certyfikat/zaświadczenie potwierdzające nabycie kompetencji będzie zawierać informacje jakie umiejętności (zestaw efektów uczenia się) nabyli uczestnicy w wyniku realizowanego projektu, oraz liczbę godzin zrealizowanych poszczególnych zajęć

Część Nr 2: (27. 2.4.) Kurs „Programista JavaScript”

Skierowany do 2 nauczycieli

Czas trwania: 160h (100 h pracy z trenerem + 60 h pracy własnej)

Termin realizacji: II - VI 2026 r.

Sposób prowadzenia: on-line

Program kursu;

1. Podstawy budowy stron – HTML, CSS, model DOM

2. Wstęp do JavaScript (VanillaJS vs inne Framework, Struktury danych, Zmienne, pętle, warunki, OOP – Programowanie Obiektowe, Semantyka standardów ES5 vs. ES6, Funkcje, klasy, obiekty, Przechowywanie danych (JSON, XML))

3. JavaScript – praktycznie (Animacje, manipulacje stroną (DOM), Tworzenie modali (okien dialogowych), Różne sposoby przechowywania danych na stronie (storages, cookies), Walidacja formularza, biblioteka jQuery oraz jQueryUI, Prosta aplikacja SPA (Single Page Application))

4. JavaScript – od kuchni (Podstawy protokołu HTTP – jak działają strony?, Czym jest DNS?, Czym jest CDN?, Przygotowanie środowiska do pracy – Node.js Jak instalować moduły? Przydatne polecenia Node.js)

5. JavaScript – piszemy frontend do prostego bloga (REST – komunikacja z backendem, Dynamiczne ładowanie treści – bez przeładowywania strony, Dodawanie nowych postów, Wyświetlanie postów, Edycja postów, Dodawanie komentarzy, Usuwanie komentarzy)

6. React (Jak zacząć?, Tworzenie nowego projektu, Struktura projektu, Jak działają komponenty?, Co składa się na komponent?, Jak działa kompilator JSX?, Routing, Tworzenie kompletnej aplikacji)

7. Wprowadzenie do kursu + przygotowanie środowiska - Konfiguracja środowiska pracy

Konfiguracja Edytora Kodu Visual Studio Code

Używanie skryptów budujących aplikację

Konfiguracja lintera, prettiera

Korzystanie z plików .env

Podstawy React.JS

Postawienie projektu w Next.JS

8. Node.JS (Mechanizmy Node.JS (stream, buffer), Moduł FS – edycja/odczyt/zapis do plików, Moduł HTTP – konfigurowanie własnego serwera, Komunikacja z bazą danych MySQL) Framework Express

9. Podstawy GIT(Pobieranie repozytoriów, Praca z gałęziami w GIT (GIT branches), Rozwiązywanie konfliktów przy łączeniu gałęzi, Git Flow i dobre praktyki korzystania z GIT)

10. Zaawansowany JavaScript (Mechanizm closure, Wykorzystanie this, Programowanie funkcyjne oraz klasowe)

11. Asynchroniczność w JavaScriptcie (Komunikacja Client-Server, Rest API, GraphQL, Uwierzytelnianie z serwerem za pomocą JWT Tokens i Access Tokens, Dobre praktyki przy korzystaniu z Promise)

12. Zaawansowany React.JS(Tworzenie komponentów w React.JS, Mechanizmy React.JS – props, state, Dobre praktyki korzystania ze stanu aplikacji – React Hooks, Kompozycja aplikacji w React.JS, Zarządzanie stanem aplikacji za pomocą React.Context, Struktura projektów Reactowych, Dobre praktyki korzystania z React.JS)

13. Praca z dodatkowymi modułami (Wprowadzenie do Jamstack, Konfiguracja i wykorzystywanie zewnętrznych bibliotek (Hubspot, FaunaDB, Firebase, Netlify forms))

14. Tworzenie aplikacji w Next.JS (Zaawansowane funkcje Next.JS – getServerSideProps, getStaticProps, getStaticPaths, API Routes za pomocą Next.JS, Kompozycja aplikacji Next.JS, Deploy aplikacji Next.JS za pomocą Vercel i Netlify, Dobre praktyki przy deployu aplikacji (integracja z GIT, env, preview))

15. Tworzenie aplikacji – Warsztaty z tworzenia aplikacji przy użyciu: Next.JS + Storyblok + Netlify, Next.JS + Firebase + Vercel

16. Szkolenie z programowania musi się odbywać w środowisku programistycznym z danego zagadnienia. Na zajęciach uczestnicy programują pod okiem osoby prowadzącej szkolenie. Osoba prowadząca powinna być praktykiem z co najmniej 5-letnim doświadczeniem w przeprowadzaniu kursów i pisaniu kodów w czasie rzeczywistym. Wprowadzenie teoretyczne musi być dostarczone przed lekcją tak aby uczestnicy mogli się z tym zapoznać. Lekcja właściwa musi być ćwiczeniem a trener – programujący w czasie rzeczywistym.

17.Szkolenie powinno odbywać się na poziomie dla nauczycieli szkoły ponadpodstawowej (mając w perspektywie przygotowanie do egzaminu info.04 technika programisty). Materiały np. zadania do wykonania powinny być dostosowane do poziomu technika programisty lub wykraczać poza jego poziom.

18. Wykształcenie prowadzącego dyplom studiów wyższych o kierunkach informatycznych lub pokrewnych np. elektroniczne, elektrotechniczne

3. Obowiązki Wykonawcy m.in.:

- Przygotowanie i przekazanie uczestnikom kursu materiałów szkoleniowych w postaci papierowej lub elektronicznej,
- Przekazanie uczestnikom zajęć materiałów teoretycznych wyczerpujących program szkolenia
- przygotowanie i przekazanie programu zajęć uwzględniający minimum: nazwę zajęć, łączną liczbę godzin, czas trwania i sposób organizacji zajęć, miejsce realizacji zajęć, imię i nazwisko osoby prowadzącej zajęcia, cele kształcenia, opis efektów uczenia się, konspekt zajęć, sposób i formę zaliczenia
- przygotowanie i przekazanie harmonogramu zajęć uwzględniający minimum: termin i miejsce realizacji zajęć, nazwę zajęć, imię i nazwisko wykładowcy i godziny realizacji zajęć. Harmonogram musi zostać uzgodniony z Zamawiającym niezwłocznie po podpisaniu umowy.

- wydanie uczestnikom szkolenia certyfikatu/zaświadczenia, będącego potwierdzeniem nabycia kompetencji uzyskanych w wyniku formalnej oceny. Certyfikat/zaświadczenie potwierdzające nabycie kompetencji będzie zawierać informacje jakie umiejętności (zestaw efektów uczenia się) nabyli uczestnicy w wyniku realizowanego projektu, oraz liczbę godzin zrealizowanych poszczególnych zajęć.

Część Nr 5: (28. 2.5.) Kurs „Tester automatyzujący”

Skierowany do 3 nauczycieli

Czas trwania: 120h (80 h z trenerem + 40 h pracy własnej)

Termin realizacji: II-VI 2026 r.

Sposób prowadzenia: on-line

Program

1. Automatyzacja- Frontend i backend, HTML, CSS(HTML atrybuty, Struktura pliku html)
2. Lokalizowanie elementów na stronie (za pomocą name, id, tagu, selektorów CSS, XPath)
3. Selenium podstawy – teoretyczne (Selenium IDE, WebDriver, Grid)
4. Środowisko programistyczne Python
5. Wprowadzenie do programowania w Python dla testerów(przechowywanie danych w zmiennych,, operacje na liczbach i tekstach, warunki logiczne, instrukcja if, listy, wybieranie elementów i fragmentów, przeglądanie danych (pętla for))
6. Narzędzia do testowania – JUnit i Selenium(Skrypt w Selenium, Zarządzanie przeglądarką za pomocą webdriver manager, Narzędzia deweloperskie w przeglądarce)
7. Tworzenie testów automatycznych aplikacji internetowych(Akcje na elementach, Weryfikacja stanu elementu, Explicit wait – WebDriverWait)
8. Framework PyTest(Pytest fixtures, Pytest parametrze)
9. Testowanie backend(Symulacja i wykonanie testu aplikacji wyszukiwarki, Page Object Pattern, Raporty z egzekucji testów)
10. Selenium Grid – utworzenie huba i node(Uruchomienie testu na Selenium Grid)
11. Środowisko programistyczne Javy(Podstawy programowania w Java, typy podstawowe, instrukcja sterującą if, switch, pętle, metody, Koncepcje programowania obiektowego)
12. Selenium – Java interakcje na elementach(Klikanie na elementy, Pobieranie tekstu, Sprawdzenie czy element istnieje na stronie, Wybieranie wartości z pola wielokrotnego wyboru, Sprawdzanie czy stan oczekiwany zgadza się ze stanem aktualnym)
13. JUNIT(Zastosowanie adnotacji: BeforeClass, AfterClass, Before, Test)
14. Teoretyczne omówienie uczestnikom szkolenia problemu w sposób usystematyzowany, analityczny przed przystąpieniem do wykonywania ćwiczeń
15. Szkolenie z programowania musi się odbywać w środowisku programistycznym z danego zagadnienia. Na zajęciach uczestnicy programują pod okiem osoby prowadzącej szkolenie. Osoba prowadząca powinna być praktykiem z co najmniej 5-letnim doświadczeniem w przeprowadzaniu kursów i pisaniu kodów w czasie rzeczywistym. Wprowadzenie teoretyczne musi być dostarczone przed lekcją tak aby uczestnicy mogli się z tym zapoznać. Lekcja właściwa musi być ćwiczeniem a trener – programujący w czasie rzeczywistym.
- 16.Szkolenie powinno odbywać się na poziomie dla nauczycieli szkoły ponadpodstawowej (mając w perspektywie przygotowanie do egzaminu info.04 technika programisty). Materiały np. zadania do wykonania powinny być dostosowane do poziomu technika programisty lub wykraczać poza jego poziom.
- 17.Wykształcenie prowadzącego dyplom studiów wyższych o kierunkach informatycznych lub pokrewnych np. elektroniczne, elektrotechniczne



Fundusze Europejskie
dla Łódzkiego



Rzeczpospolita
Polska

Dofinansowane przez
Unię Europejską



Projekt pn.: „**Kompetentny zawodowiec – satysfakcja dla siebie i innych**” współfinansowany ze środków Europejskiego Funduszu Społecznego Plus w ramach Programu Regionalnego Fundusze Europejskie dla Łódzkiego na lata 2021-2027, FELD.08.08-IZ.00-0059/23-00

Powiat Łowicki, Zespół Szkół Ponadpodstawowych Nr 2 Centrum Kształcenia Zawodowego im. Tadeusza Kościuszki w Łowiczu, ul. Blich 10, 99-400 Łowicz
www.zsp2lowicz.pl, adres e-mail: stazprzepustka@zsp2lowicz.pl

3. Obowiązki Wykonawcy m.in.:

- Przygotowanie i przekazanie uczestnikom kursu materiałów szkoleniowych w postaci papierowej lub elektronicznej,
- Przekazanie uczestnikom zajęć materiałów teoretycznych wyczerpujących program szkolenia - przygotowanie i przekazanie programu zajęć uwzględniający minimum: nazwę zajęć, łączną liczbę godzin, czas trwania i sposób organizacji zajęć, miejsce realizacji zajęć, imię i nazwisko osoby prowadzącej zajęcia, cele kształcenia, opis efektów uczenia się, konspekt zajęć, sposób i formę zaliczenia - przygotowanie i przekazanie harmonogramu zajęć uwzględniający minimum: termin i miejsce realizacji zajęć, nazwę zajęć, imię i nazwisko wykładowcy i godziny realizacji zajęć. Harmonogram musi zostać uzgodniony z Zamawiającym niezwłocznie po podpisaniu umowy. - wydanie uczestnikom szkolenia certyfikatu/zaświadczenia, będącego potwierdzeniem nabycia kompetencji uzyskanych w wyniku formalnej oceny. Certyfikat/zaświadczenie potwierdzające nabycie kompetencji będzie zawierać informacje jakie umiejętności (zestaw efektów uczenia się) nabyli uczestnicy w wyniku realizowanego projektu, oraz liczbę godzin zrealizowanych poszczególnych zajęć.