

# Programista JavaScript z wykorzystaniem AI

STUDIA PODYPLOMOWE



## Program zajęć

**9**

**178**

**11**

**2**

Liczba miesięcy nauki   Liczba godzin zajęć   Liczba zjazdów   Liczba semestrów

### **Wprowadzenie do Web Developmentu (32 godz.)**

- Type Script: Składnia, zmienne, typy danych, funkcje, interfejsy, klasy, enumy itd.
- JSON: Struktura danych i przetwarzanie w formacie JSON.
- Ecma Script: Najnowsze standardy.
- Preprocessor SCSS: Tworzenie i zarządzanie stylami w aplikacjach webowych.
- Responsive Web Design (RWD): Projektowanie responsywnych stron internetowych.
- Debugowanie i korzystanie z konsoli developerskiej

### **Wprowadzenie do Angulara (32 godz.)**

- Moduły Angulara: Tworzenie i zarządzanie modułami aplikacji (do wersji 14 od 14 komponenty standalone)
- Komponenty: Projektowanie i implementacja komponentów.
- Serwisy: czym są serwisy oraz przypadki użycia.
- Dyrektywy: Wykorzystywanie wbudowanych i tworzenie własnych dyrektyw.
- Pipe'y: Wykorzystanie wbudowanych i tworzenie własnych pipe'ów.
- Szablony HTML i interpolacja: Tworzenie interfejsu użytkownika.
- Bindowanie danych: Przekazywanie i wyświetlanie danych w komponentach.
- Wstrzykiwanie zależności (Dependency Injection)

### **Zaawansowane konstrukcje Angulara (40 godz.)**

- Serwisy http: Implementacja i wykorzystanie usług w Angularze.
- Routing i guardy: Konfiguracja routingu i nawigacji w aplikacji.
- Formularze: Tworzenie formularzy i obsługa danych wejściowych.
- Observable i RxJs: wprowadzenie do reaktywnego programowania.
- Mockowanie danych: przygotowanie aplikacji do pracy bez backendu.

### **Testowanie w Angularze (8 godz.)**

- Jednostkowe testy: Testowanie komponentów i usług.



- Narzędzia do testowania: Wybór i konfiguracja narzędzi do testowania.

## **Wprowadzenie do komercyjnego programowania (24 godz.)**

- Praca w zespole: Efektywna współpraca w grupie programistycznej.
- Narzędzia śledzenia pracy: Wykorzystanie narzędzi takich jak Jira do zarządzania projektami.
- Procesy w rozwoju oprogramowania: Zrozumienie czym jest Story, Task, Bug, Refinement, wycena.
- Kryteria akceptacji: Jak określić, kiedy zadanie jest uznane za ukończone.

## **Code Review i dobre praktyki (8 godz.)**

- Code Review: Proces przeglądu kodu i analiza zmian.
- GitHub: Tworzenie repozytoriów, zarządzanie zmianami i pull requestami.
- Gitflow: Praktyki pracy z gałęziami w repozytorium.
- Standardy kodu: Przyjęte standardy i dobre praktyki programistyczne.

## **Praktyczne zastosowanie Angulara (32 godz.)**

- Tworzenie rzeczywistych projektów: Implementacja aplikacji przy wykorzystaniu Angulara w ramach
- pracy zaliczeniowej z możliwością konsultacji i przeglądu kodu.

## **Projekt (2 godz.)**

- Projekt zaliczeniowy

## **Forma zaliczenia**

- Praktyczny projekt końcowy pisany indywidualnie lub grupowo pod opieką prowadzącego