

# Zasady zaliczenia przedmiotu TIN

## Studia dzienne

---

### Spis treści

---

- Zasady ogólne
- Oddawanie zadań
- Progi ocen
- Harmonogram zajęć
- Projekt zaliczeniowy

### Zasady ogólne

---

1. W semestrze można otrzymać max. 100 punktów:
  - 60 punktów - zadania domowe
  - 40 punktów - projekt
2. Zaliczenie ćwiczeń od 50%.
3. Dozwolone 15% nieusprawiedliwionych nieobecności na zajęciach.

### Oddawanie zadań

---

- Wykonane zadania należy umieścić w odpowiednim folderze w systemie GAKKO w terminie ustalonym na zajęciach.
- Zadania są sprawdzane indywidualnie w czasie zajęć.
- Nie przewiduje się poprawiania zadań domowych, umieszczona wersja na GAKKO powinna być finalna.

## Progi ocen

Procent	Ocena
<50%	2
50% - 59%	3
60% - 69%	3,5
70% - 79%	4
80% - 89%	4,5
>=90%	5

## Harmonogram zajęć

Nr	Data	Temat
1	01.10.2024	HTML
2	08.10.2024	CSS
3	15.10.2024	JavaScript - wprowadzenie
4	22.10.2024	JavaScript - programowanie w środowisku przeglądarki
5	29.10.2023	Node.js - wprowadzenie
6	05.11.2024	Node.js / Express - budowa aplikacji WWW
7	12.11.2024	Node.js / Express - wzorzec MVC i trwałości danych
8	19.11.2024	React.js - wprowadzenie
9	26.11.2024	React.js - Single Page Application
10	03.12.2024	Service Oriented Architecture
11	10.12.2024	Projekt zaliczeniowy - ustalenie tematu
12	17.12.2024	Projekt zaliczeniowy - prezentacja statycznej wersji interfejsu
13	07.01.2025	Projekt zaliczeniowy - konsultacje
14	14.01.2025	Projekt zaliczeniowy - obrona (1/2)
15	21.01.2025	Projekt zaliczeniowy - obrona (2/2)

# Projekt zaliczeniowy

---

Wskazana punktacja to maksymalna możliwa do uzyskania ilość punktów; w wypadku niespełnienia/częściowego spełnienia wymagań ocena będzie obniżona.

Zbuduj aplikację WWW obejmującą funkcjonalność po stronie serwerowej i przeglądarkowej. Technologią domyślną jest Node.js oraz popularne frameworki na nim oparte oraz JavaScript po stronie przeglądarki. Zastosowanie innych technologii jest dopuszczalne w przypadku indywidualnego uzgodnienia z prowadzącymi. Aplikacja powinna spełniać następujące wymagania (wymagania określone dla każdej oceny, aby otrzymać ocenę wyższą należy również spełnić wszystkie wymagania dla ocen niższych):

## (poziom-1: od 0 do 20 punktów):

---

Co najmniej 3 połączone tabele bazy danych (w tym jedna relacja wiele do wielu z co najmniej jedną dodatkową kolumną oprócz kluczy obcych w tabeli łączącej)

Co najmniej 3 różne typy kolumn

Co najmniej 2 kolumny w każdej tabeli (oprócz klucza podstawowego)

Domyślnie wszystkie dane tych trzech wymagalnych encji z ww. schematu powinny być edytowalne poprzez funkcjonalność stworzonej aplikacji WWW. Od tej reguły można odstąpić, jeśli uzasadnia to logika biznesowa, a analogiczne elementy funkcjonalności zaimplementowano dla którejś / którychś z pozostałych encji

Funkcjonalności:

- Dodawanie, aktualizacja, usuwanie rekordów do bazy danych
- Wyświetlanie listy wszystkich rekordów dla każdej tabeli (tylko najważniejsze kolumny)
- Wyświetlanie widoków szczegółowych (wszystkie kolumny + rekordy połączone relacjami)
- Walidacja danych po stronie klienta i serwera

Ograniczenia:

- Żadnych projektów „zrobiłem to do pracy”.
- Zakaz kopiowania istniejących tutoriali
- Zakaz używania generatorów typu scaffold
- MySQL lub MongoDB jako baza danych, projekt musi zawierać łatwe do wykonania instrukcje dotyczące aktualizacji informacji o połączeniu do bazy danych

Rozwiązanie musi zawierać:

- Kod aplikacji frontendowej i backendowej
- Skrypt tworzenia bazy danych
- Skrypt bazy danych do tworzenia przykładowych danych

- Wszelkie skrypty kompilacji itp. użyte do zbudowania aplikacji
- Krótki dokument opisowy, obejmujący następujące dane:
  - wykaz zaimplementowanej funkcjonalności ze wskazaniem, czy dana funkcjonalność jest przedmiotem kontroli dostępu (np. wymaga zalogowanego użytkownika; dowolnego lub mającego przydzieloną określoną rolę);
  - wykaz kroków niezbędnych do uruchomienia aplikacji z wykorzystaniem przesłanego kodu (np. czy aplikacja opiera się na danym systemie baz danych, a jeśli tak, jaka wersja i ustawienia konfiguracyjne muszą być zapewnione);

Uwaga! Przygotowując makiety interfejsu użytkownika (pośredni krok w trakcie realizacji projektu zaliczeniowego, nadsyłany w postaci zbioru stron HTML), upewnij się, że Twoje rozwiązanie:

- czytelnie przedstawia koncepcję funkcjonalności przewidzianej do zrealizowania (np. czy określony element danych ma być w danym widoku możliwy do edytowania),
- łącznie reprezentuje widoki odpowiadające funkcjonalności odpowiadającej co najmniej kompletnie kryteriów poziomu-1 z powyższej listy.

## (poziom-2: od 0 do 30 punktów):

---

Dodatkowe funkcjonalności:

- Rejestracja i logowanie (w tym niezbędne tabele bazy danych)
- Różne funkcjonalności w zależności od statusu użytkownika (zalogowany/gość) – np. tylko zalogowani użytkownicy mogą modyfikować dane
- Paginacja wyświetlanych list

## (poziom-3: od 0 do 40 punktów):

---

Dodatkowe wymagania:

- Aplikacja frontendowa powinna być zbudowana jako Single Page Application

Dodatkowe funkcjonalności:

- Co najmniej 3 role użytkowników (np. zalogowany, gość, administrator) z różnymi dostępnymi funkcjonalnościami
- Wprowadzenie uprawnień na poziomie zasobu (np. klient może przeglądać tylko własne zamówienia, menadżer może przeglądać i aktualizować tylko dane pracowników ze swojego działu)
- Umiędzynarodowienie (co najmniej 2 języki)