

# Wymagania i opis zaliczenie projektu dla przedmiotu 'Podstawy tworzenia aplikacji w oparciu o usługi Azure'

---

<b>PROJEKT .....</b>	<b>3</b>
FORMA.....	3
FORMALNOŚCI .....	3
OCENIANIE .....	4
PRZYKŁADOWE PROJEKTY .....	5
Portal do oceniania książek/filmów.....	5
Portal do uproszczonego zarządzania klubem fitness.....	5
Bardzo uproszczony sklep internetowy .....	5

# PROJEKT

## FORMA

- **zaliczenie i oddanie projektu odbywa się w formie dema**
- dema wszystkich projektów będą się odbywały jednego dnia
- każdy projekt będzie miał taki sam czas na przeprowadzenie dema
- każdy student będzie miał możliwość rezerwacji slotu czasowego w którym chce przeprowadzić demo
- **dema powinno pokrywać następujące zagadnienia:**
  - o funkcjonalność aplikacji
  - o architektura aplikacji z naciskiem na komponenty Azure które zostały użyte wraz z uzasadnieniem ich użycia
  - o wszelkie inne dodatkowe rozważania na temat zaimplementowanego rozwiązania jak:
    - metryki zbierane przez aplikację,
    - prezentację kosztów itp.
- w demie ze strony Relativity będą uczestniczyły dwie osoby
- w trakcie dema reprezentanci Relativity mogą/będą zadawać pytania dotyczące projektu oraz zasobów Azure

## FORMALNOŚCI

- **źródła projektu powinny zostać udostępnione oceniającym co najmniej tydzień przed dniem dema, tj. - w zależności od wybranego terminu – do 16 stycznia 2024 lub 23 stycznia 2024**
- **źródła można udostępnić w postaci linku do GitHub, BitBucket, lub spakowanego archiwum .zip**
- **zgłoszenia na poszczególne terminy dema oraz źródła należy wysyłać na uzgodniony adres email: [building\\_in\\_azure@relativity.com](mailto:building_in_azure@relativity.com)**
- **w skład źródeł projektu powinien wchodzić dokument, który prezentuje wizualnie architekturę aplikacji ze szczególnym uwzględnieniem użytych komponentów Azure oraz wizualizacją interakcji między nimi**

## WYMAGANIA TECHNICZNE

- cała infrastruktura użyta w projekcie może być stworzona ręcznie w Azure Portal
- rdzeniem projektu powinna być aplikacja wdrożona jako Azure WebAPP
- aplikacja może zostać napisana w dowolnym języku programowania
- Proces budowania aplikacji powinien być zautomatyzowany
- Kod aplikacji powinien być przetestowany jednostkowo a testy te powinny być uruchamiane każdorazowo w procesie budowania aplikacji
- powstała aplikacja powinna realizować dowolną funkcjonalność biznesową

- funkcjonalność biznesowa nie będzie oceniana
- Interfejs użytkownika aplikacji nie będzie oceniany
- konfiguracja aplikacji powinna być przechowywana w ApplicationSettings
- wdrożenie nowej wersji aplikacji powinno odbywać się bezpiecznie (aplikacja może być wdrażana zarówno na środowisko testowe jak i na produkcyjne)
- w projekcie powinna zostać zaimplementowana dowolna forma dbania o bezpieczeństwo (autoryzacja/autentykacja)
- w aplikacji powinien zostać użyty jakikolwiek trwały storage (SQL, Cosmos, Blob)
- w aplikacji powinna zostać zaimplementowana przynajmniej jedna z funkcjonalności monitoringu
  - o alerty - np. alertowanie o krytycznych błędach w aplikacji
  - o logowanie - integracja logowania w aplikacji z Azure Log Analytics
  - o metryki - np. w formie metryk biznesowych z działania aplikacji
  - o application insights
- wymagania dodatkowe, mogące podnieść ocenę:
  - o w aplikacji pewne wymagania mogą być realizowane przez Azure Functions
  - o aplikacja może się składać z więcej niż z jednego komponentu Azure (np. dwie Azure WebApp) a komunikacja między nimi może odbywać się przez Azure Event Grid lub Azure Service Bus

## OCENIANIE

**Oceną wyjściową jest ocena 3.0, którą otrzymać można za zaimplementowanie minimalnych wymagań stawianych projektowi dotyczących użytych komponentów Azure:**

- Azure WebApp
- Storage (SQL lub Blob)
- Wdrażanie oraz budowanie aplikacji z użyciem Azure DevOps
- Autoryzacja/autentykacja
- Jedna z funkcjonalności monitoringu: alerty/logowanie/metryki/application insights

Użycie dodatkowych komponentów Azure podnosi ocenę według poniższych reguł:

- CosmosDB -> +0.5
- Azure Function -> + 0.5
- Każda dodatkowa funkcjonalność monitoringu (alerty/logowanie/metryki/application insights) -> +0.5
- EventGrid -> +0.5
- Message Bus -> + 0.5

Zatem na ocenę 5.0 przykładowo wymagane jest użycie następujących komponentów Azure w projekcie:

- Azure WebApp
- Storage (SQL lub Blob)
- Deploy z użyciem Azure DevOps
- Autoryzacja/autentykacja

- Trzy z funkcjonalności monitoringu: alerty/logowanie/metryki/application insights
- Azure Functions
- CosmosDB

## PRZYKŁADOWE PROJEKTY

### Portal do oceniania książek/filmów

Przykładowe funkcjonalności dla portalu:

- Zakładanie konta użytkownika
- Logowanie się jako użytkownik
- Dodawanie/usuwanie książek/filmów (obrazek, tytuł, itp.)
- Ocenianie książek/filmów

### Portal do uproszczonego zarządzania klubem fitness

Przykładowe funkcjonalności dla portalu:

- Logowanie jako administrator
- Dodawanie zajęć jako administrator
- Blokowanie użytkowników (np. za nieodwołane zajęcia) jako administrator
- Zakładanie konta jako użytkownik
- Logowanie się jako użytkownik
- Zapisywanie się/odwoływanie zapisu na zajęcia jako użytkownik
- Zbieranie punktów za odbyte zajęcia jako użytkownik
- Wymiana punktów na nagrody jako użytkownik

### Bardzo uproszczony sklep internetowy

Przykładowe funkcjonalności dla sklepu:

- Zakładanie konta jako użytkownik
- Logowanie się jako użytkownik
- Dodawanie/usuwanie przedmiotów z koszyka
- Składanie zamówienia jako użytkownik
- Otrzymywanie notyfikacji jako użytkownik
- Zbieranie punktów za zakupy jako użytkownik
- Wymiana punktów na nagrody jako użytkownik