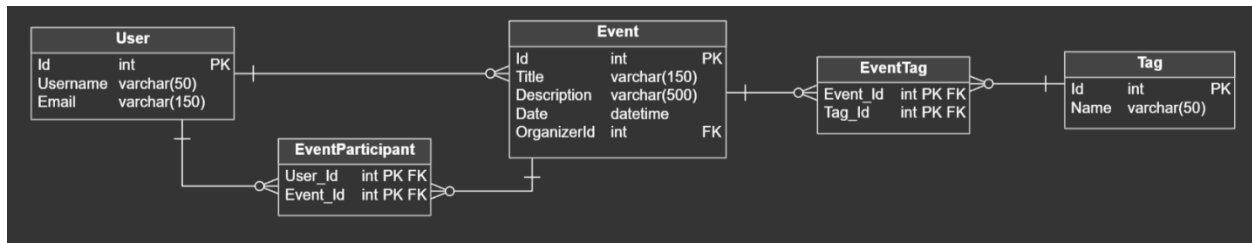


Czas: 1h30m

Liczba punktów do zdobycia: 26

Korzystając z technologii .NET oraz EFCore, utwórz aplikację typu WebApi, której zadaniem będzie obsługa poniższej bazy danych. Skorzystaj z dostarczonego skryptu sql oraz podejścia DataBaseFirst, aby wygenerować odpowiednie klasy do pracy z entity frameworkiem.



Zadanie 1.

Utwórz **endpoint HTTP GET** pod adresem `/api/events/details`, który zwróci szczegółowe informacje o wszystkich wydarzeniach.

Zwracane dane powinny zawierać:

- Tytuł i opis wydarzenia
- Datę Wydarzenia
- Nazwę organizatora
- Listę uczestników (ich nazwy użytkownika)
- Listę tagów przypisanych do wydarzenia

💡 *Upewnij się, że odpowiedź to lista obiektów DTO – nie wystawiaj encji bezpośrednio.*

Przykład odpowiedzi od serwera:

```
[
{
  "id": 1,
  "title": "Hackathon 2025",
  "description": "24h programowania i pizzy",
  "date": "2025-07-10T09:00:00",
  "organizer": {
    "id": 3,
    "username": "jan_kowalski"
  }
}
```

```

    },
    "participants": [
      { "id": 4, "username": "anna_nowak" },
      { "id": 5, "username": "robert_lewandowski" }
    ],
    "tags": [
      { "id": 1, "name": "programming" },
      { "id": 3, "name": "freeFood" }
    ]
  },
  {
    "id": 2,
    "title": "Spotkanie zarządu",
    "description": "Omówienie planów Q4",
    "date": "2025-07-12T16:00:00",
    "organizer": {
      "id": 2,
      "username": "marta_boss"
    },
    "participants": [],
    "tags": [
      { "id": 2, "name": "business" }
    ]
  }
]

```

Zadanie 2.

Utwórz **endpoint HTTP PUT** pod adresem `/api/events/{id}`, który umożliwi edycję wydarzenia o podanym id.

Zaktualizować należy:

- Tytuł i opis wydarzenia
- Listę tagów (czyli zapisać nowe powiązania w tabeli EventTags, usuwając stare)
- Listę uczestników (czyli zapisać nowe powiązania w tabeli EventParticipants, usuwając stare)

Zakładamy, że użytkownik frontendowy wysyła pełny zestaw danych (czyli **nowy stan listy tagów i uczestników**), a API aktualizuje odpowiednio obie tabele asocjacyjne.

Przykład ciała zapytania:







```
{
  "title": "Hackathon 2025 – Edycja rozszerzona",
  "description": "48h kodowania, noclegi i jeszcze więcej pizzy!",
  "date": "2025-07-10T09:00:00",
  "tagIds": [1, 4, 5],
  "participantIds": [4, 5, 6]
}
```

Przykład ciała odpowiedzi:

```
{
  "message": "Wydarzenie zaktualizowane pomyślnie.",
  "eventId": 1,
  "updatedTags": [1, 4, 5],
  "updatedParticipants": [4, 5, 6]
}
```

Co wolno studentowi w trakcie kolokwium?

Student może:

-  korzystać z template'u aplikacji, bo nie ma sensu wynajdywać WebAPI na nowo (pamiętajcie tylko o tym, żeby usunąć "ściągawki" z ostatecznego rozwiązania - za każdy pozostawiony śmieć lecą punkty),
-  sięgać do własnych notatek, wcześniej wykonanych prac oraz świętych zwojów z wykładów,
-  przeszukiwać dokumentację internetową oraz relikty wiedzy z portali takich jak StackOverflow (o ile nie klika w magiczne guziki z napisem "AI"),
-  słuchać muzyki w słuchawkach, dopóki nie przeradza się to w prywatny koncert symfoniczny,
-  poprosić prowadzącego o wyjaśnienie treści zadania — ale nie, nie chodzi tu o "czy może Pan mi to zrobić?",
-  zgłosić problem techniczny, który nie podlega ocenie — np. .NET odmawia współpracy, Visual Studio się buntuje etc.

✗ Czego robić nie wolno (nawet jeśli wygląda kusząco):

Student nie może:

- 🗣️ komunikować się z innymi osobami,
- 📄 wymieniać się kodem — wykrycie plagiatu skutkuje wyzerowaniem wszystkich podobnych do siebie prac,
- 🤖 korzystać z pomocy sztucznej inteligencji.

🕒 Kolokwium musi:

- ⌚ zostać oddane **do końca zajęć** — każda sekunda po terminie liczy się jako zamach na punktację (czyt. *drastyczne obniżenie oceny*),
- 📁 zostać **wrzucone na odpowiednie repozytorium na GitHubie** — nie Google Drive, nie zip w e-mailu, nie wiadomość na Messengerze,
- 🔑 **kompilować się bez błędów** — brak zaliczenia

🔔 Błędy kardynalne (i ich skutki):

Występek	Kara
brak .gitignore	-10%
wrzucenie śmieci do repozytorium (np. /bin, /obj, IDE, lunchbox)	-30%
łamanie konwencji nazewniczych	-10%
brak podziału na warstwy (minimum: kontroler i serwis)	-25%

projekt niekompiluje się (sprawdzić, czy na githubie znajdują się wszystkie potrzebne pliki, przed oddaniem pracy)	Kolowkium niezaliczone
Nieodpowiednie kody http	-5% per kod