Accelist 3+1 Training   
Competency Examination

.NET INTERMEDIATE

# Rules

* Open book. You may not ask any questions, and NO DISCUSSIONS!!!
* Each point is worth 5 score.
* When grading a point, if ***Point Killer*** is discovered, that point score WILL BE ZERO!
* **C# Coding Conventions:**
  + PascalCase on class names and class members (fields, properties, methods, etc.) names.
  + camelCase on local variable (var) names and method parameter names.
  + Not **await**-ing an asynchronous method, which includes blocking it using: **.Result, .Wait()**
* **Accelist Coding Conventions:**
  + All control structure code blocks (e.g. **if-else, foreach-in**) must be written using braces {}
  + Use **LINQ Method Syntax** (Lambda Expression / Q => …) instead of **Query Syntax** (from …), except when **join**-ing tables.
* Exam duration is one day (**8 hours**).
* **Clone** then later **Push** your work to the following Git repository: [accelist/ExamNETIntermediate: For Exam .NET Intermediate 2024 (github.com)](https://github.com/accelist/ExamNETIntermediate) | https://github.com/accelist/ExamNETIntermediate
  + Create a branch with the format: NIM\_Name

# Point Killer

1. **Coding Conventions violations.**
2. All date-time units **(DATETIME2, DATETIMEOFFSET)** sent to the API must be in UTC time zone!
   * C# 🡪 **DateTime.UtcNow, DateTimeOffset.UtcNow**
3. Synchronously calling the API and parsing the results instead of asynchronously.
   * Using the methods without async.

# Good Luck!

Buatlah sebuah aplikasi manajemen metadata music dengan fitur sebagai berikut:

1. Pada saat aplikasi dimulai, tarik data genre dan data lagu dari API yang disediakan.
   * Endpoint-endpoint API yang tersedia dilampirkan di dokumen ini.
2. Sebuah ListBox yang menampilkan judul-judul lagu yang ada.
   * Ketika salah satu item dipilih, maka inputan lagu akan diupdate dengan data lagu tersebut.
3. Button agar user bisa merefresh list lagu dengan data baru dari API.
   * Data song list yang ada direset dan digantikan dengan data baru dari API.
4. Button agar user bisa memilih untuk menambahkan lagu baru. Ketika akan menambahkan lagu baru, pilihan lagu di ListBox akan dibersihkan dan inputan lagu akan dibersihkan.
5. Sebuah TableLayoutPanel yang berisi inputan untuk lagu, serta tiga Button untuk Add, Update, dan Delete:
   * Ketika ada item yang dipilih dari ListBox, inputan ini akan berisikan detail dari lagu yang dipilih.
   * Inputan ini memiliki field sebagai berikut:
     + Title: TextBox
     + Artist: TextBox
     + Genre: ComboBox, pilihan yang tersedia diambil dari API genre.
     + Length: dua NumericUpDown, untuk masing-masing inputan
     + IsAvailable: CheckBox
     + Button Add: menambahkan lagu
     + Button Update: mengupdate lagu yang dipilih
     + Button Delete: menghapus lagu yang dipilih
   * Inputan harus divalidasi sesuai kriteria berikut:
     + Title:
       - Tidak boleh kosong
     + Artist:
       - Tidak boleh kosong
     + Genre:
       - Tidak boleh kosong
       - Pilihan harus ada di data genre dari API.
     + Length:
       - Terbagi menjadi dua inputan, untuk menit dan untuk detik
       - Menit dan detik tidak boleh kosong.
       - Detik harus bernilai dari 0 sampai 59.
     + Release date:
       - Tidak boleh kosong
       - Tanggal maksimal adalah 7 hari ke depan.
     + Is Available:
       - Tidak boleh kosong
   * Ketika input ada yang tidak valid, tampilkan Message Box berisi error yang ada.
   * Ketika tidak ada item yang dipilih di ListBox, pilihan yang tersedia adalah untuk menambahkan lagu baru.
     + Ketika user menambahkan lagu baru, panggil API untuk menambahkan lagu baru, dan tampilkan MessageBox success ketika berhasil.
   * Ketika ada item yang dipilih di ListBox, pilihan yang tersedia adalah untuk mengupdate dan menghapus lagu.
     + Ketika user mengupdate lagu, panggil API untuk mengupdate lagu, dan tampilkan MessageBox success ketika berhasil.
     + Ketika user menghapus lagu, panggil API untuk menghapus lagu, dan tampilkan MessageBox success ketika berhasil.
   * Setelah operasi Add, Update, atau Delete, refresh kembali daftar lagu dari API.
6. Searching
   * Terdapat TextBox yang berisi keyword untuk pencarian. Pada saat TextBox terisi, maka pencarian dilakukan berdasarkan isi dari TextBox tersebut. Pada saat TextBox tidak terisi, semua data akan ditampilkan.
   * Search dilakukan terhadap list lagu yang ada berdasarkan Title atau Artist dari lagu.
   * Search dilakukan dalam aplikasi, e.g. client-side.

# Bonus Scores

1. Add XML / JSDOC Documentations to every class and class members (properties, methods, fields) you write!
   * C#: type **///** above class / class members, fill in its description (what it is / does) between the **<summary>** tag.

# Troubleshooting

1. I cannot install NuGet package, help!
   * Try running **dotnet restore** from the project folder, then **dotnet build**.
2. Visual Studio failed to run the project and gives out weird error, help!
   * Try running **dotnet restore** from the project folder, then **dotnet build**. (Then **dotnet run** if web app)
   * Alternatively, try **Build 🡪 Clean** Solution, then **CTRL + F5**.
3. Google is your best friend.

# Attachment: API documentation

Base URL: https://new-dev.accelist.com:10000

* GET /api/genre

Sample return:

A screenshot of a computer program

Description automatically generated

* GET /api/song

Sample return:

A screenshot of a computer code

Description automatically generated\

* POST /api/song/create

Body: JSON object

A screenshot of a computer code

Description automatically generated

* PUT /api/song/{id}

Parameter:

* + id: ID dari lagu

Body: JSON object

A screenshot of a computer code

Description automatically generated

* DELETE /api/song/{id}

Parameter:

* + id: ID dari lagu