Dear

Thank you for applying for the backend development position at Seamless Digital.

As part of the process, we would like to assess your knowledge and development skills with a small technical challenge. Please find the details below, and feel free to ask any questions if something is unclear.

Instructions:

- There are three parts. Do not feel pressure to spend loads of time on your responses but showing a good technical understanding is important.
- Your submission can be in whatever format is best for you. An email reply, a link to a GitHub repository, or a .zip file containing your answer are all fine.
- Please include comments, or separately include an explanation of any coding decisions you've made.
- Feel free to use generative AI for Part 1! However, for Part 2 and 3, please give your own opinions and answers.

Part 1: Practical Challenge

Task: Build a Sample API

Create a simple To-Do backend with APIs, using a local SQL database.

- Fetch the To-Do list from https://dummyjson.com/todos and store it.
- Stored To-Do items along with the following additional fields:
    - Category (optional, relation to category list)
    - Priority (1 to 5; 1 means top priority, 5 means low priority, default is 3)
    - Location (optional, latitude & longitude)
    - Due date (optional, Datetime)

- Default storage of **Category List:**
    - Title (String)
    - Parent category (option, key)

- Client needs a new REST API which will:
    - Support CRUD operations for a To-Do item.
    - Combine current weather information and a To-Do item if the Location is set.
        - Weather data can be obtained from: https://www.weatherapi.com/
        - Return only the current temperature and current condition in text (e.g. Sunny, Partly Cloudy, etc.)

- Create a C# RSET API that supports the following operations:
    - Add To-Do item and generates unique ID.
    - Update To-Do item.
    - Delete a To-Do item by ID.
    - Search To-Do item(s) by title or by priority or by due date.

- Provide at least 1 unit test.

Part 2: Theory Questions

Please provide a short-written response to the following questions:

1. What issues can arise from using `async void` methods in C#? When should you use `async Task` instead, and why?

2. Explain the potential problems with using `static` variables in a multi-threaded or web application context. How can this affect application behavior?

3. What is the difference between the == operator and the `.Equals()` method in C# when comparing objects?

4. What happens if you do not implement the `IDisposable` interface for a class that uses unmanaged resources. What are the consequences, and how can you prevent them?

5. Explain the key differences between Common Table Expressions (CTEs) and temporary tables. In which scenarios would you prefer using one over the other?

6. Do you see any issues with this LINQ query?

```
public class Book
{
    public int Id { get; set; }
    public string Title { get; set; }
}

var books = dbContext.Books
    .Where(p => IsRecommended(p.Title))
    .ToList();

bool IsRecommended(string title)
{
    return title.StartsWith("A") && title.EndsWith("Z");
}
```

Part 3: Opinion Question

What is your stance on exception handing in backend applications, and how do you typically implement it in your C# projects?

Thank you and looking forward to reading your response.