C#/Js Fullstack exercise

Client-Server Application

Objective

Your task is to create a simple client-server application. The client should be able to upload a text file with data in the format described below which the server will process, save and return the data parsed in JSON format.
The data must be saved in the server in any format (on drive, streamed and added to a text file, in a file DB, or a DB instance of your choice) but in the returned JSON data an ID must be added to identify the new registry.
If any error occurred during the processing the server must return an error message with the format described below. The client should display the error message in a friendly format (ex, alert)
The processed data is returned to the client which displays the results in table format. As a bonus, the data can be displayed in a bar graph. For this you can use your own implementation of HTML5 Canvas or any existing library.
Also as a bonus, the server can include 2 extra methods: one to retrieve a registry by ID and another to delete a registry by ID. For these methods, the client can provide an input field to indicate the RegistryID and 2 buttons to trigger the Get and Delete actions.
All code must be uploaded to a repository like github, bitbucket, etc.

Code must be documented in English (JSDocs, C# Summaries, or any of your choice)

Estimated time to solve, approx 8 hours

Tasks

- **Client**
    - Upload a file
    - Display the server results
    - Display any error messages
- **Server**
    - Parse the uploaded file
    - Returns clear messages in case of error
    - Save the data in the server
    - Return the parsed data in JSON format with a new ID field

Bonus Tasks

- **Client**
    - Validating the data on file (*label: text/number, value: number, color: hex*)
    - Retrieve a registry by ID
    - Delete a registry by ID
    - Display the results in a Bar graph.
- **Server**
    - Method to retrieve a registry by ID
    - Method to delete a registry by ID

Implementation

The client side should be implemented as a SPA using Vanilla JS or any JS-framework/library of your choice (simple HTML/Ajax, Angular, React, Svelte, Alpine.js, etc)
The server side should be implemented in C# as a simple .Net Framework or .NetCore application.
For both sides you can implement the pattern of your choice.

Data samples

File

The file is a simple Txt file where each row (line) represents a bar.
Every bar should have a name/label (letters and numbers), a value (numbers only) and a color (hex format) separated by a “#” (hashtag) char (hex value).

Example

The following example represents 3 bars: 1 for Argentina with a value of 1500 and a red color, 1 for Brazil with a value of 4005 and a green color and 1 for Colombia with a value of 345 and blue color.

```
Argentina#1500#FF0000
Brazil#4005#00FF00
Colombia#345#0000FF
```

Returned data

The return data must be in JSON format and should implement the following class interface

- **Data**
    - ID: (int) auto generated ID
    - Count: (int) amount of rows/bars in the file
    - Timestamp: (datetime) of the procedure
    - Rows: (List<Row>) process rows/bars
- **Row**
    - Name: (string) title/label of the bar
    - Value: (decimal) value of the bar
    - Color: (string) color of the bar in hex

Example

```
{
        "ID": 123,
        "Count": 3,
        "Timestamp": "2023-08-11 00:32:14",
        "Rows":[
                {
                        "Name": "Argentina",
                        "Value": 1500,
                        "Color": "FF0000"
                },
{
                        "Name": "Brazil",
                        "Value": 4005,
                        "Color": "00FF00"
                },
                {
                        "Name": "Colombia",
                        "Value": 345,
                        "Color": "0000FF"
                }

        ]
}
```