

## Task 1: Create a REST API with ASP.NET Core for scheduling reminders via email or Telegram

### Objective:

- Create a web application that allows users to schedule reminders via email or Telegram using a REST API built with ASP.NET Core and PostgreSQL (or MySQL, MSSQL or any other database of your choice)

### Requirements

- The application should have CRUD (create, read, update, delete) endpoints for managing reminders in the database.
- The "create" endpoint should accept the following parameters: "to" (string), "content" (string), "sendAt" (datetime), and "method" (string).
  - The "to" parameter should be a valid email address for email reminders, or a chat ID for Telegram reminders.
  - The "sendAt" parameter must be greater than the current date and time. Users can't schedule for past time.
  - The "method" parameter should be either "email" or "telegram", indicating which method to use for the reminder.
- The "read" endpoint should return a list of all reminders in the database.
- The "update" endpoint should allow updating the "to", "content", "sendAt", and "method" parameters of a reminder using its ID.
- The "delete" endpoint should allow bulk deleting of reminders from the database using IDs
- The application should send emails using any email service or mock service of your choice.
- The application should send Telegram messages via the Telegram Bot API.
- The application should send reminders via the specified method when their sendAt time has come.
- The application should be well-structured, with clear separation of concerns, proper exception handling, and appropriate use of design patterns.
- The code should be written in C# and follow the Microsoft C# coding conventions.

### Timeline

- You have 2 days to complete this task.

### Evaluation

- We will evaluate your code based on correctness, readability, maintainability, and adherence to best practices.
- We will also consider your design choices, error handling, and testing strategy.
- Bonus points for implementing additional features such as authentication, rate limiting, and validation.

### Example of how this API could work in steps

1. Create a reminder via the "create" endpoint by sending a POST request with the following JSON payload:

```
{
  "to": "1399964683",
  "content": "Don't forget to attend the meeting at 2:00 PM today.",
  "sendAt": "2023-05-12T14:00:00Z",
  "method": "telegram"
}
```

- This creates a reminder to be sent via Telegram to the user with the chat ID "139996483" at 2:00 PM UTC on May 12, 2023.
2. The reminder is stored in the database.
3. At the scheduled time (2:00 PM UTC in this example), the application retrieves the reminder from the database and sends it via Telegram to the specified user.
4. The user receives the reminder message on their Telegram account.

Similarly, you can create a reminder via email by setting the "method" parameter to "email" and providing a valid email address in the "to" parameter. When the reminder is due, the application sends an email containing the reminder message to the specified email address.