What to build:

- “Imagegram” - a system that allows you to upload images and comment on them
- no frontend/UI is required

User stories (where the user is an API consumer):

- As a user, I should be able to create posts with images (1 post - 1 image)
- As a user, I should be able to set a text caption when I create a post
- As a user, I should be able to comment on a post
- As a user, I should be able to delete a comment (created by me) from a post
- As a user, I should be able to get the list of all posts along with the last 2 comments to each post

Functional requirements:

- RESTful Web API (JSON)
- Maximum image size - 100MB
- Allowed image formats: .png, .jpg, .bmp.
- Save uploaded images in the original format
- Convert uploaded images to .jpg format and resize to 600x600
- Serve images only in .jpg format
- Posts should be sorted by the number of comments (desc)
- Retrieve posts via a cursor-based pagination

Non-functional requirements:

- Maximum response time for any API call except uploading image files - 50 ms
- Minimum throughput handled by the system - 100 RPS
- Users have a slow and unstable internet connection

Usage forecast:

- 1k uploaded images per 1h
- 100k new comments per 1h

Preferable tools:

- .NET (C# or F#) / Node.js / Go / Python
- Azure Functions / AWS Lambda / k8s (or similar cloud managed services)
- GitHub

Expected result:

- Source code
- Whatever you think is necessary to deliver a prototype/MVP of this project
- Details about what should be done in order to ship it on production

Notes:

- You are recommended to spend no more than 6 hours on this challenge
- You don’t have to complete all the stories - the goal is to show your strengths
- Focus on good design and clean implementation
- Develop your system applying the best software development practices
- The non-functional requirements and the usage forecast should be only considered as design guidelines - there is no need to prove it
- If you have any questions feel free to ask