Large Attachments Handling

# Task

Describe the web forms support for large attachments (~100MB). Describe in text how to handle storage and downloads from the submissions list, considering thousands of submissions with multiple attachments each. Include architecture, data structure, and REST API.

# Architecture

**Storage**

I would consider 2 points

* Cloud-based Object Storage: like Azure Blob Storage or analogues uses direct streaming to the object storage and can successfully store large files. This solution can be scaled horizontally
* Normal Database SQL or NoSQL for storing metadata and submission details, references URL, file name, etc

**Backend (API)**

REST API will stream file directly without caching. We will consider also:

1. **Error Handling & Timeout Handling**: Appropriate logging errors and try-catch to avoid code from just crashing. Timeout settings on both the client and server sides to prevent issues during large uploads.
2. **Chunking**: file can be splitted into several pieces. Possible network failure might cause the corruption of "big" type of file.
3. **Rate Limiting & Throttling:** Implement rate limiting and throttling mechanisms to prevent the server from being overwhelmed by too many large file uploads simultaneously.
4. **Scheduling Job**: We can schedule the uploading using the Hangfire library. It allows us to monitor, track progress, enque jobs and the retry if needed. UI will be fast and responsive.

# Data Structure

I would add a data about this file to the main model. I represent it as a JSON:

{

"id": "guid",

//all normal submission fields (see Domain class library)

"attachments": [

{

"fileId": "guid",

"fileName": "string",

"fileType": "string", // file type (e.g., image/jpeg, application/pdf)

"fileSize": "int", // Size in bytes

"fileUrl": "string", // URL of the file in cloud storage (e.g., S3 URL)

"uploadTimestamp": "timestamp"

}

]

}