Analysis & Design

CoreThree – Technical Test

# Feeds:

|  |  |
| --- | --- |
| Input | [Requirements](Requirments.docx). <Research.docx> |
| Output | [Solution](file:///D:\Data\Family\Terry\Jobs\Ash%20Ali%20and%20Nasim%20Ghouri%20CoreThree\Solution) |
| Output | [Visual Studio Solution](../Solution/CoreThreeAPI/CoreThree.sln) |

# R01: Your task is to create a service using .NET that:

R01.1 queries

or

R01.2 is triggered from an AWS S3 bucket for any new files.

Set up a rest API project in Visual studio 2022 .Net 6 called CoreThreeAPI

# R02: It should load them and write them to a database (preferably MongoDB).

Two parts to this

Design the poco class for the Json data that would be entered in a Mongo Db – 1 Json string

For now, I will use a local Mongo Db

# R03: Additional features:

Lower Priority

You can add in any additional features to the service as you see fit (e.g., sending an email on each file upload).

Suggest: if we do it then make it the email feature.

# R04: Create an API with at least two endpoints.

These endpoints shall provide the following functionality:

* R04.1: GET details of all file uploads.
* R04.2: GET details of a specific file upload

## Solution:

Implement the 5 basic CRUD ops: Add, Update, Delete, Get, Get All

# R05: non-functional:

You can use this exercise to display any technical skills that you wish to demonstrate.

## Solution:

All part of the service sir!

# R06: Bonus Task 1: use

# Focus the files, service, and API around a ticketing domain.

## Solution:

Analyze the typical detailed use cases and likely information needed on a ticketing form

# R07: Bonus Task 2: Create a basic frontend web app

which allows a user to upload files to the above mentioned S3 bucket and allow the display of information from the API calls.

## Solution:

Implement a basic mvc app or razor pages and swagger