.NET developer (Full Stack) developer test

Candidate Name: ____Ahmed Thabet_____

Test Date **:** ____23/8_____

Test Start Time **:** ____Friday 1:00 PM_____

Test End Time **:** ____Sunday 1:00 PM_____

Tasks count : _____1_____

TASK: Pictures gallery via Azure Storage

Introduction

In this task you will work with Azure storage to build a pictures gallery web application.

You will use Azure storage as the backend repository for storing the images. You will use Angular as your (frontend) presentation layer.

The web application is required to communicate with a Web APIs layer that encapsulates all the system functionality (all CRUD operations).

For more information about Azure storage ***blobs*** read this article:
https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction

Here is an example of how to use Azure storage blobs with .NET:
https://docs.microsoft.com/en-us/azure/storage/blobs/storage-dotnet-how-to-use-blobs

For this task you will use the following storage account information to be able to connect to Azure storage programmatically:

Storage Account name: fgtest3
Storage Account Key:
ePycbjNKmeHKeBvK2T5F9AuUPRAj2d0s9irsMXGaoUYEElbL9yw91uElEZW2yrJp/t5tUCRcFc5Sqs/a/fUtGw==

Storage Account Connection string:
DefaultEndpointsProtocol=https;AccountName=fgtest3;AccountKey=ePycbjNKmeHKeBvK2T5F9AuUPRAj2d0s9irsMXGaoUYEElbL9yw91uElEZW2yrJp/t5tUCRcFc5Sqs/a/fUtGw==;EndpointSuffix=core.windows.net

A storage container (aka folder) called “images” is created under the storage account “fgtest3” listed above.

That container allows anonymous read for all contained blobs (aka files). Here is a sample image uploaded to it:

https://fgtest3.blob.core.windows.net/images/SMI_Education_Logo.png

Requirements

1- The backend:

Create Web APIs using ASP.NET to be used to upload, list and delete images from the Azure storage account listed above. These web apis will leverage the Azure Storage SDK to communicate directly with the Azure cloud storage.

Create an additional Web API controller with an action that allows deleting all images uploaded to the application.

2- The frontend:

Use Angular to build a web application that will use the APIs implemented in the previous step.

The web application should do the following tasks:

1- Upload a new image
2- List all uploaded images
3- Download an image
4- Delete an uploaded image
5- Deleting all uploaded images (this shouldn't be allowed by anonymous users).

Delivery

How to deliver your work

Create the required code projects for the task and push them to a public GitHub repo and email the link after you are done. Adding documentation is a big plus.

This task in general will test the following:

1- Following .NET coding best practices
2- Understanding on .NET Web API and MVC
3- Understanding of web development in general
4- Experience with Angular
5- Ability to learn new technologies like Azure storage

How to achieve the best results

1- Completing the task start to finish on time has a high mark. Try to finish all the tasks as much as you can first, then you can work on optimization\refactor\enhancements if needed later.
2- Provide working solutions that compile successfully without any errors
3- Make sure you commit or upload to your repo all the required assets for your solution to compile successfully
4- Write code to be easily readable
5- Avoid over engineering your solution to show off some patterns or competencies that may not be needed for this task. You are measured on the code quality not complexity
6- Simple is always better than complex
7- Write code comments as much as you can that explains what your code is trying to do
8- Try to write code with performance in mind as much as you can

GOOD LUCK