Jessica Nguyen

CSS 436, Program 4

Contents

[URL LOCATIONS 1](#_Toc65011440)

[DESIGN IMPLEMENTATION 1](#_Toc65011441)

[PERFORMANCE REPORT 3](#_Toc65011442)

[How the site will scale with load 3](#_Toc65011443)

[How monitoring is done on the site 4](#_Toc65011444)

[Estimate of SLA 4](#_Toc65011445)

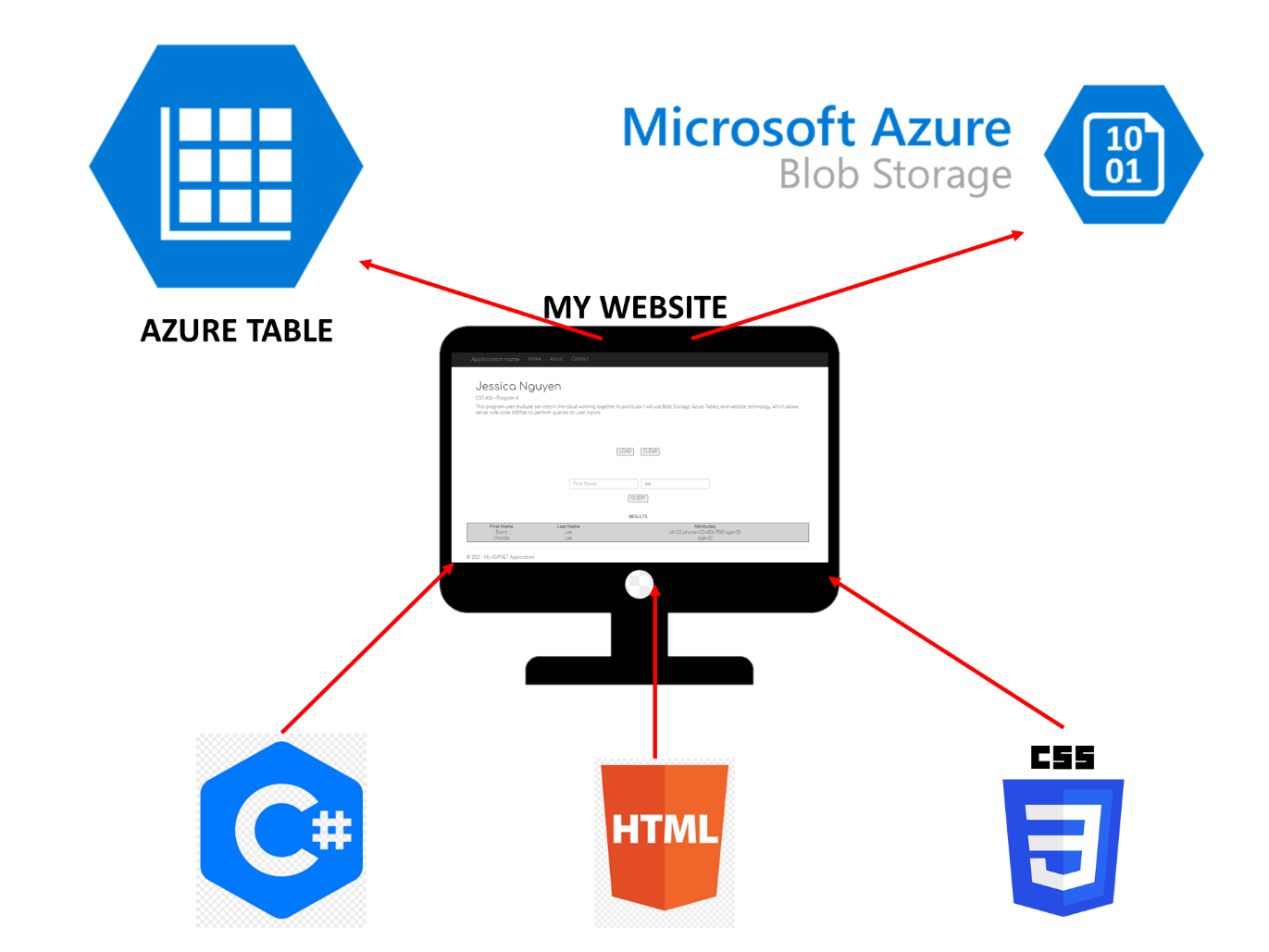
[SAMPLE WEBSITE OUTPUT 5](#_Toc65011446)

## URL LOCATIONS

My website: <https://nguyen-program4.azurewebsites.net/>

My Blob object: <https://proram4.blob.core.windows.net/object/input.txt>

## DESIGN IMPLEMENTATION



**🡨AZURE APP SERVICE**

1. Azure Cloud services used:
   1. Azure Web Services to host my web app
   2. Azure Blob Storage to store my text file
   3. Azure Table to store Person objects as NoSQL database
2. Back-end technology used:
   1. Visual Studio ASP.NET MVC Web Application with .NET Framework 4.7.2
   2. Default.aspx.cs file to code my program behavior in C#
   3. Default.aspx file to code my program component layout in HTML
   4. Site.css to code my program styling in CSS
3. Program components:
   1. **class Person : TableEntity** -- This is a class representing a Person as an Azure TableEntity for adding into Azure Table on the cloud

Each Person has a string fname, string lname, and string attributes parsed from text input

* 1. **public partial class \_Default : Page** -- The main class for all functionalities

**IConfigurationRoot GetConfiguration()** -- sets up the credentials without hard-coding into the cs file

**Button1\_Click(object sender, EventArgs e)** -- LOAD BUTTON

* precondition: All credentials are properly set up
* postcondition: program will load input Blob object into my own Blob Storage,

parse the Blob text file and add the list of people in the text as key/value attributes in my own Azure Table instance

**CopyBlob()** -- copies predetermined input Blob into my own Blob Storage account

* precondition: input Blob URL is valid and public
* postcondition: Azure Blob Storage copies a separate Blob instance into my account
* If the original Blob text file is changed, the Blob instance will be updated accordingly

**ConnectToBlob()** -- instantiate a BlobBaseClient object to interact with Azure Blob Storage Service

* precondition: credentials are set up in appsetting.json

**ConnectToTable()** -- instantiate a CloudTableClient object to interact with Azure Table Service

* precondition: credentials are set up in appsetting.json

**CreateTable()** -- create a table instance on my Azure Table account

**NOTE: After pressing CLEAR button, wait at least 5-10 seconds before clicking on LOAD button -- when you CLEAR the entities, it takes a while to parse through the entire table depending on the size. If you try to LOAD again too fast, you will have to wait until my backoff logic has successfully reloaded the command.**

**Button2\_Click1(object sender, EventArgs e)** -- CLEAR BUTTON

* postcondition: deletes my Blob file on Blob Storage;
* deletes my whole Table with all key/value pairs on Azure Table;
* clears out the text input boxes in the html file

**Deserialize()** -- parses the input text file and separate out first name, last name, and attributes and add them to Azure Table

* precondition: credentials are set up in appsetting.json

**AddToTable(Person p)** -- helper method to add Person object to Azure Table

* precondition: Person input is not null object
* postcondition: if the fname and lname are same, table will override the old value and update with the one thats added the latest;
* return false if Person object passed in is null

**Button3\_Click1(object sender, EventArgs e)** -- QUERY BUTTON

* precondition: returns on the output with matches of the user input in the Table database
* postcondition: if user enters in only first name, all objects with matching first name keys will be displayed;
* if user enters in only last name, all objects matching last name keys will be displayed;
* if user enters in both first and last name, one key/value object matching both will be displayed

**Format(string s)** -- formats all input text to capitalize only the first letter of input string

* postcondition: returns empty string if input is null or only whitespace

## PERFORMANCE REPORT

### How the site will scale with load

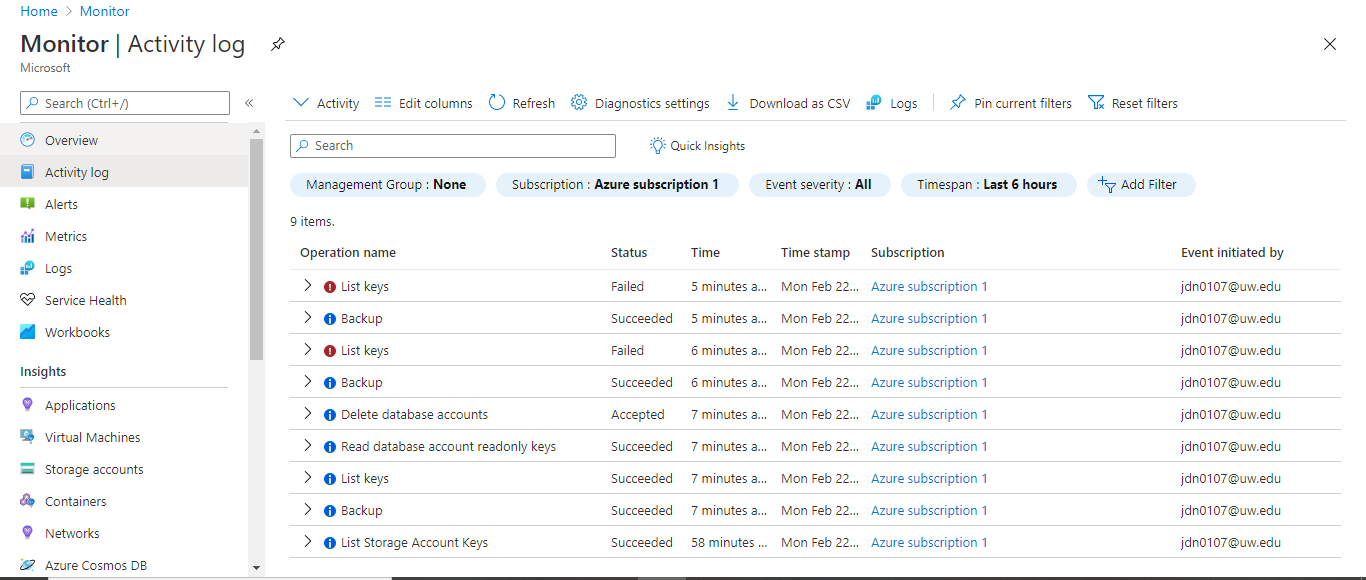
* Since the website is hosted on Azure Web Service, the hosting (availability) will be done by Microsoft, which will automatically scale for the request. The web server or OS can easily cache pages in memory
* My website is hosted on the [Free Plan](https://docs.microsoft.com/en-us/azure/app-service/web-sites-monitor#howtoviewusage), so it has a quota on 60 minutes CPU time a day. After using for an hour, it will return an error to the client. Unfortunately, if the quota is used up, please try again the next day
* [Azure Blob Storage](https://docs.microsoft.com/en-us/azure/storage/blobs/scalability-targets) can handle up to 100 MiB a block, and if the size exceeds that, the data will be split into multiple blocks that can handle up to 4.75 TiB, with up to 500 requests per second. If the application reaches the limit workload, Azure Storage returns error code 503 (Server Busy), my application uses an exponential backoff policy for retries
* [Azure Table](https://docs.microsoft.com/en-us/rest/api/storageservices/designing-a-scalable-partitioning-strategy-for-azure-table-storage#aserrd) has a scalability target of 500 entities per second. Load balancing occurs at the Distributed File System (DFS) layer. When load balancing occurs, the partition becomes offline for a few seconds, and Azure automatically creates more instances behind the scenes to run the request. My application uses an exponential backoff policy for retries

### How monitoring is done on the site

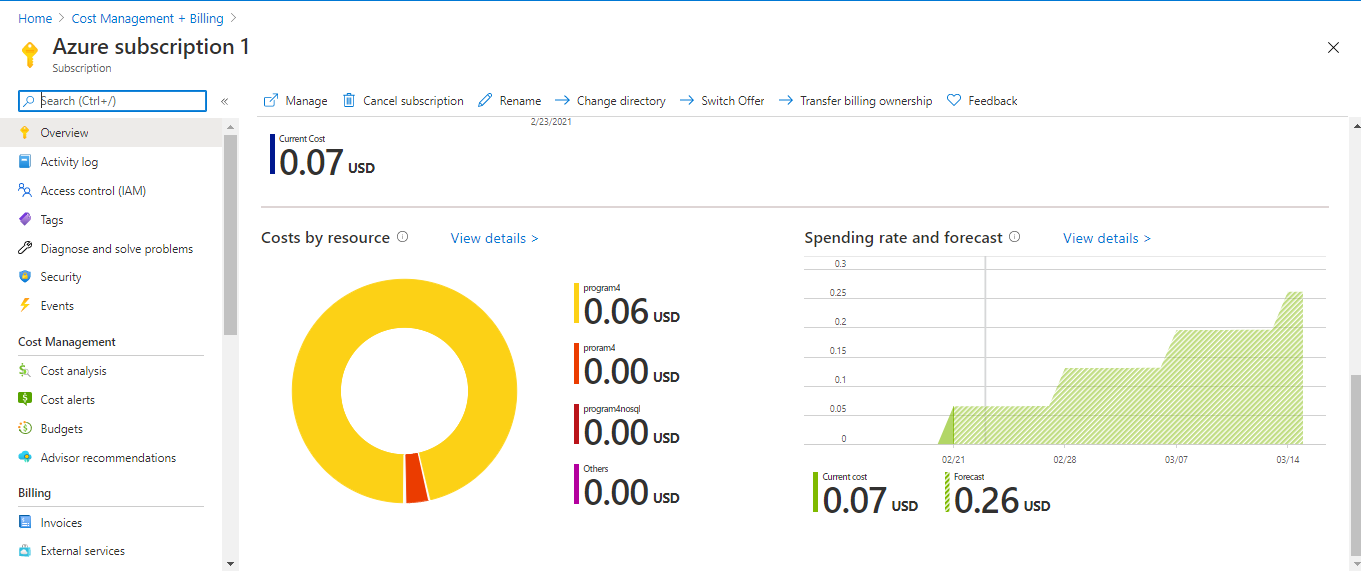
For all Cloud Services, I set the budget to be $1 a month and will get notified if the usage exceeds 80% of the budget.

The Activity Log section shows all past actions on cloud services such as OperationName, Status, Time and who made the commands. This way, I can monitor or track what I did that didn’t work or worked when I am testing and deploying my web app.

A sample log image:



Cost management metrics screen:



### Estimate of SLA

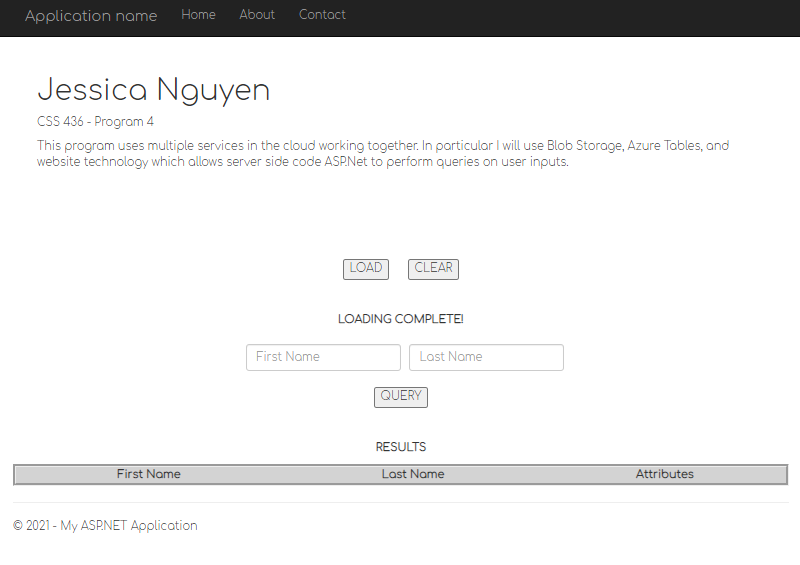
Link: <https://azure.microsoft.com/en-us/support/legal/sla/>

For Azure Web Services and Storage, they promise a 10% service credit back on the bill if the Monthly Uptime Percentage is less than the guaranteed SLA, and 25% credit if less than 99% availability overall.

* Azure Web Services: 99.95%
* Azure Storage Accounts - Blob Storage: 99.99%
* Azure Storage Accounts - Table: 99.99%
* Total SLA: 0.995 \* 0.9999\*0.9999 = 0.9993 **= 99.93%**

## SAMPLE WEBSITE OUTPUT

Finish loading screen:



Query screen:

