CSE 6331 Cloud Computing
Summer 2018, © DL, UTA, 2018

Programming Assignment 1
Introduction to Cloud Storage and UI
Due: In Blackboard

Task: You are will create a cloud-based picture and associated information storage and retrieval system with a (local) web interface (UI)

Description:
One of the most common uses of "Clouds", is shared or backup storage. SaaS, with a friendly interface.

Your assignment is to provide a local interface to a cloud service that you will implement that will allow a user to upload a meta-information table "people.csv", a .csv (text) table followed by several individual pictures. Then the user may do queries that select some (or none) pictures, specified in the people table.
For example :

Name	Grade	Room	Telnum	Picture	Keywords
Nora	100	550	1000010	nora.jpg	Nora is nice
Jees	98	420		jees.jpg	Jees is Jees
Abhishek	98			abhishek.jpg	Abhishek is not Jees
Dave	40	525	-0		Who is this

Which will look like (in the "people.csv"):

```
Nora,100,550,1000010,nora.jpg,Nora is nice
Jees,98,420,,jees.jpg,Jees is Jees
Abhishek,98,,,abhishek.jpg,Abhishek is not Jees
…
```

And your cloud-based "service" will allow a user to:
+ Search for Nora (Name) and show her picture on a web page.
+ Search for (display) all pictures where the grade is less than 99.
+ Add a picture for Dave
+ Remove Dave
+ Change Jees keywords to "Jees is still Jees"
+ Change Abhishek's grade
And similar…

You may use any reasonable (non-hardcoded) implementation of the people table: Hashes, a SQL (or non-) table, or even a dictionary or array.
Pictures are binary entities stored on the cloud provider storage, in any manner you wish (files, DB tables, hashes, etc.).

You should handle conditions such as: missing data (fields, attributes), unavailable pictures, attempts to upload the same named picture twice, pictures that are of incorrect type ("nora.txt"), and similar.

Please, submit in Blackboard. Work must be individualized, but may be done in a group.

You must submit this lab, working (or partially) by the due date.
Your program should be well commented and documented, make sure the first few lines of your program contain your name, this course number, and the lab name and number.
Your comments should reflect your design and issues in your implementation.
Your design and implementation should address error conditions.