Exam 1 – 12 hours to complete

Development of REST API services with Python, mySQL and Flask

The main objective of this assignment is to create several API endpoints that can be used in other applications to perform CRUD operations on a dataset revolving about watch collecting.

1. Set up a table in your database named 'watches' according to the following schema:

id make model type purchaseprice saleprice

Pick appropriate types for each column. The id of every record should be created automatically. The make represents the manufacturer of the watch. The model should represent the watch model number or model name. The type should indicate what type of watch it is. For instance, there are 'mechanical', 'automatic', 'quartz', 'solar', 'digital', etc. The purchaseprice should list for how much the watch was bought. The saleprice needs to list for how much a particular watch has been sold. Note: This column needs to allow NULL, because a watch doesn't necessarily need to be sold (yet). If a watch is still in possession of the collector, this field is null.

Make sure before submission you have at least 5 records in your table. You might have to do some research if you are not familiar with the subject. For instance, one entry could look like this:

Id	make	model	type	purchaseprice	saleprice
1	Hublot	Big Bang Meca	chronograph	31000	44000

2. Create the following 4 REST API service endpoints to perform CRUD operations:

1. /api/watch method GET: This endpoint should allow to get a list of all watches, sorted by the profit. Unsold watches should be last in the list.
2. /api/watch method POST: This endpoint should allow the caller to add a new watch.
3. /api/watch method DELETE: This endpoint should allow the caller to delete a watch from the table, if they are authorized to so so via a token. The universally accepted master token for this operation is: 880088
4. /api/watch method PUT: This endpoint should allow the caller to update the saleprice of an existing watch (no other fields).

This exam requires you to have a remote MySQL database setup in AWS or another cloud of your choice. (local DB setups will not be accepted and result in zero credit for this portion of the exam).

It is recommended that you use MySQL Workbench for creating the table and adding the 5 initial records. But you can use any tool that you are comfortable with. You do not have to submit any SQL code that fills the table.

Other Requirements

• Make sure the username and password for the database you setup are credentials you're willing to share. Do not use personal passwords for this exam, which you might be using anywhere else.

• Make sure your project compiles and runs. If the code doesn't run, you will forfeit points. This includes submitting all files required by your project!

Extra credit

Use UUID as data type for the autogenerated ID column of your table.

What to Turn in

Submit your source code (all Python files) as single zip file via Blackboard. Your code has to have meaningful comments explaining how you worked through the problem. You can use the sql.py and the creds.py files we have created in class, but don't forget to submit them as part of your source code.

Grading rubric for this exam

Item	Points
Endpoint 1	20
Endpoint 2	20
Endpoint 3	20
Endpoint 4	20
Code comments	5
Database and table setup with records	15
Extra Credit	10
Total	100
Deduction 50% if code doesn't compile	(up to -50)