Integrity through Non-Fungible Assessments in Cloud-Based Technology Courses

Aspen Olmsted, Ph.D.
Fisher College
aolmsted@fisher.edu

About Author

- 25 years in Industry Developing N-Tier Business Solutions
- Program Director and Professor at Fisher College in Boston
- Acting Program Director for New York University Cyber Fellows Program
- Research is in N-Tier and Cloud Application Correctness

Agenda

- Scalable Assessments
- Headed vs Headless Autograders
- Programming Labs - Prerequisite Knowledge
- Coding Course Labs - Database and Operating Systems Examples
- Tool Course Labs - Networking Examples
- Derivate Labs
- Scripting Synchronization
- Peer Review
- Derivative Quiz Pools
- Dividing Up Work
- Programming Links
- Questions

Scalable Assessments

Our goal is to have assessments in the online courses that provide integrity while not requiring large number of TAs to support the assessments

- Autograded Labs
- Peer Review Projects
- Derivative Quiz Pools

Current Ratios

- Cyber Fellows - 1 Teacher/100 Students/2 TAs
- Prerequisite Knowledge - 0 Teachers/100 Students/0 TAs
- edX - 1 Teacher/750 Students/0 TAs

Headed Auto-graders

In Computer Science, Information Technology and Cybersecurity our inclination is to create autograded labs that simulate the operating system or application experience.

Headed Grader Platforms:

- Codio - www.codio.com
- Vocareum - https://www.vocareum.com/
- Practice Labs - https://www.practice-labs.com/

Headless Auto-graders

Gradescope provides a docker container that allows students to upload file and scripts run and return the results to the student

- Programming Labs - Students upload their code (programs, classes or functions) and they are unit tested
- Tool Labs - Students upload a file demonstrating their use of tool (i.e. pcap file)

Prerequisite Knowledge Programming Labs

Adult students may have gained their programming prerequisite knowledge a long time ago. In Cyber Fellows we provide lectures and labs with auto-graders for

- Python
- C++
- Data Structures
- x86 Assembler (Future)

Example C++ Lab

- ## BMI Metric Function

Description

Body mass index (BMI) is a number calculated from a person's weight and height. According to the Centers for Disease Control and Prevention, the BMI is a fairly reliable indicator of body fatness for most people. BMI does not measure body fat directly, but research has shown that BMI correlates to direct measures of body fat, such as underwater weighing and dual-energy X-ray absorptiometry. The formula for BMI is: Weight/Height^2

Where weight is in kilograms and height is in meters.

The name of the function should be named bmimetricf and the function should take two parameters which are the weight in kilograms and the height in meters to use to calculate the BMI.

Example C++ Lab Continued

Here is an example call to the function

```
int weight = 50;
float height = 1.58;
printf(“BMI is: %3.2f”,bmimetricf(weight, height)) ;
```

The output of the code above would be: BMI is: 20.03

File Name

bmimetricf.cpp

Function Signature

```
float bmimetricf(int weight, float height);
```

Score

There are five tests each worth 2 points

Course Programming Labs

Similar labs have been used in the following courses

- Operating Systems Course - Students upload function implementations in C - Stub compiled and linked - Students given one test case in documentation and code has several test cases
- Database Course - Students upload SQL queries. A python executer takes the query and sends to a database

Tool Labs

In Networking, Network Security and Penetration Testing students use tools such as Wireshark, TCPDump, etc

- Questions about the lab file - In Networking Students fill out Google Form that answers questions about the pcap file. They upload the pcap file and the answers are compared (show demo)
- Parse File - A parser can be written that scores the students from the output file

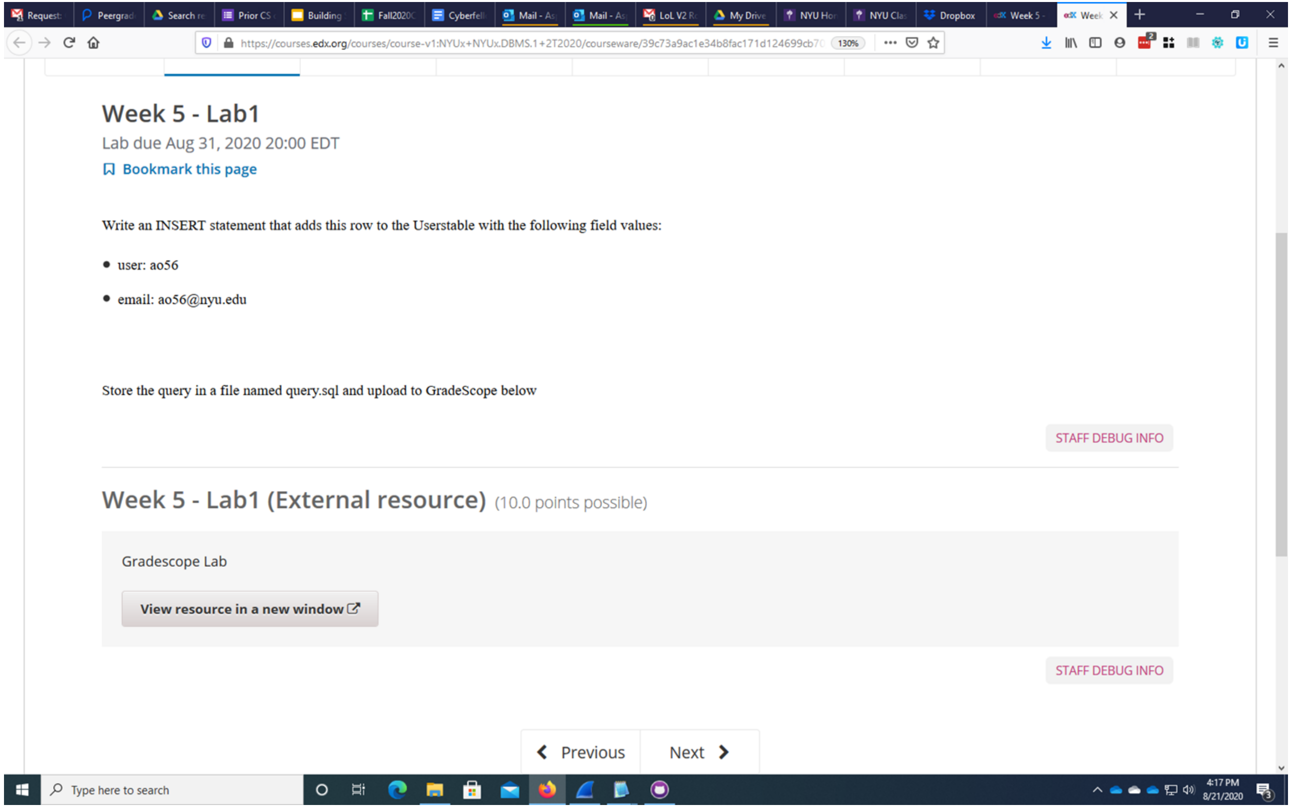
Derivate Labs

A derivative lab changes the lab so each student requires a different solution

- Utilize email, location, time to change the question and required solution
- For tools lab - send student to a site that requires they identify themself so the user is in the pcap or output file.

This increases learning because you can encourage discussion instead of trying to dissuade discussion

Example Derivate Lab on edX



https://courses.edx.org/courses/course-v1:NYUx+NYUx.DBMS.1+2T2020/courseware/39c73a9ac1e34b8fac171d124699cb7
130%
Type here to search
4:17 PM
8/21/2020

Week 5 - Lab1

Lab due Aug 31, 2020 20:00 EDT

Bookmark this page

Write an INSERT statement that adds this row to the Userstable with the following field values:

- user: ao56
- email: ao56@nyu.edu

Store the query in a file named query.sql and upload to GradeScope below

STAFF DEBUG INFO

Week 5 - Lab1 (External resource) (10.0 points possible)

Gradescope Lab

View resource in a new window

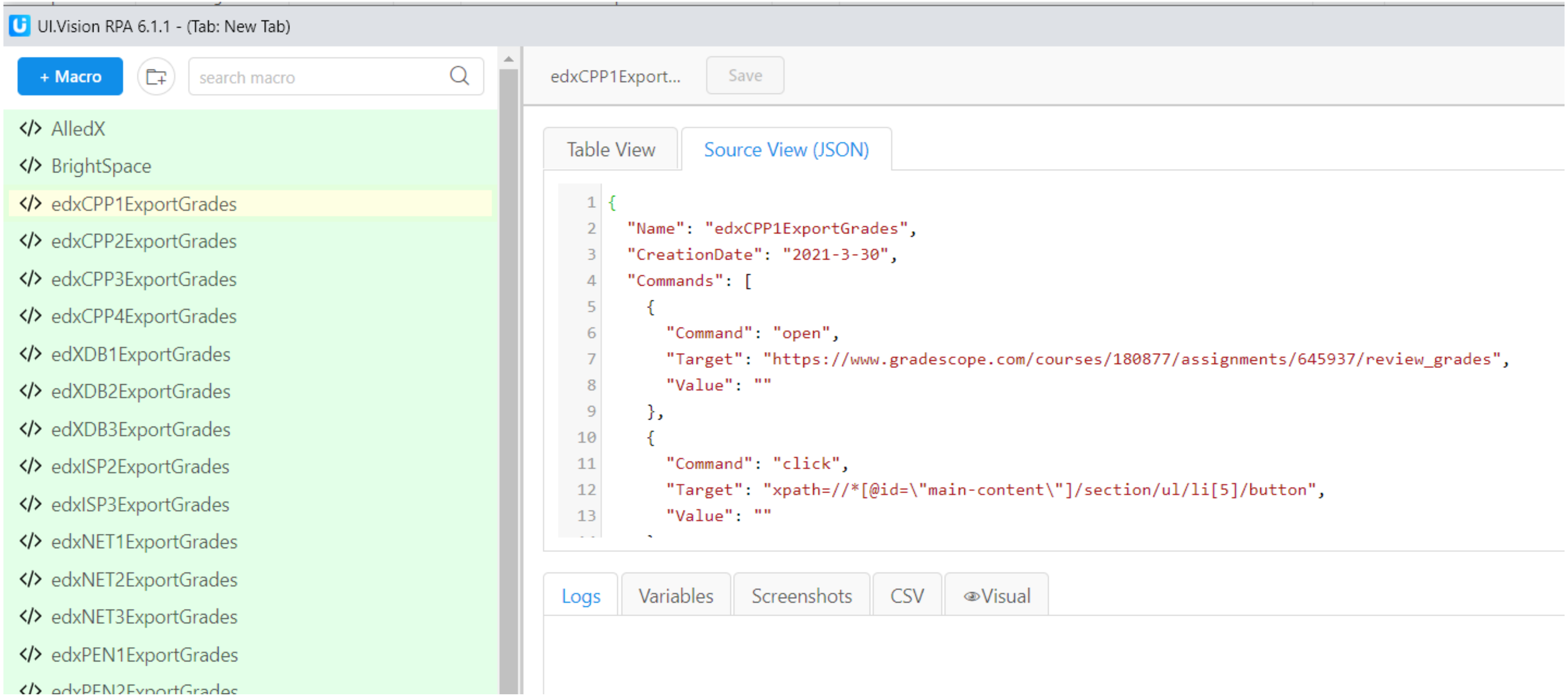
STAFF DEBUG INFO

Previous Next

Scripting Grade Synchronization

- Gradescope matches roster based on email
- Allows for self-service account merge
- Push Grades from Gradescope to Sakaii/edX
- Script from UI.Vision RPA/iMacros/headless Chrome

Scripting Grade Synchronization (cont)



- AlledX
- BrightSpace
- edxCPP1ExportGrades
- edxCPP2ExportGrades
- edxCPP3ExportGrades
- edxCPP4ExportGrades
- edXDB1ExportGrades
- edXDB2ExportGrades
- edXDB3ExportGrades
- edxISP2ExportGrades
- edxISP3ExportGrades
- edxNET1ExportGrades
- edxNET2ExportGrades
- edxNET3ExportGrades
- edxPEN1ExportGrades
- edxPEN2ExportGrades

```
{
  "Name": "edxCPP1ExportGrades",
  "CreationDate": "2021-3-30",
  "Commands": [
    {
      "Command": "open",
      "Target": "https://www.gradescope.com/courses/180877/assignments/645937/review_grades",
      "Value": ""
    },
    {
      "Command": "click",
      "Target": "xpath=//*[@id=\"main-content\"]/section/ul/li[5]/button",
      "Value": ""
```

Questions?

Send to aolmsted@fisher.edu