**User Manual**

* Firstly download the uploaded files from the canvas.
* There will be two folders name Phase 1 and Phase 2, which will contain five files all together, 2 files in phase 1 (two tiers), 3 files in Phase 2 (three tiers).

**Requirements:**

* If you do not have base Python IDE (IDLE), or any other IDE that can run python codes, please download it beforehand.
* Find the base python IDE download through this link: https://www.python.org/downloads/
* The client and server files can be separated into 2-3 machines and all the machines should be **connected to the same network** in order for them to work successfully.

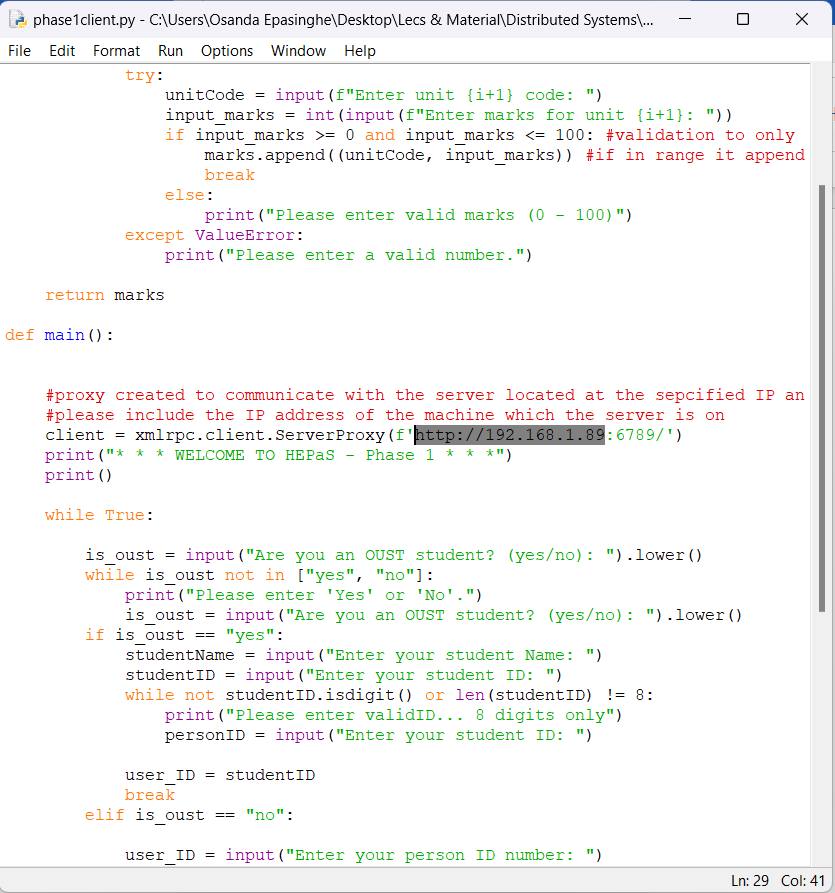
**After downloading:**

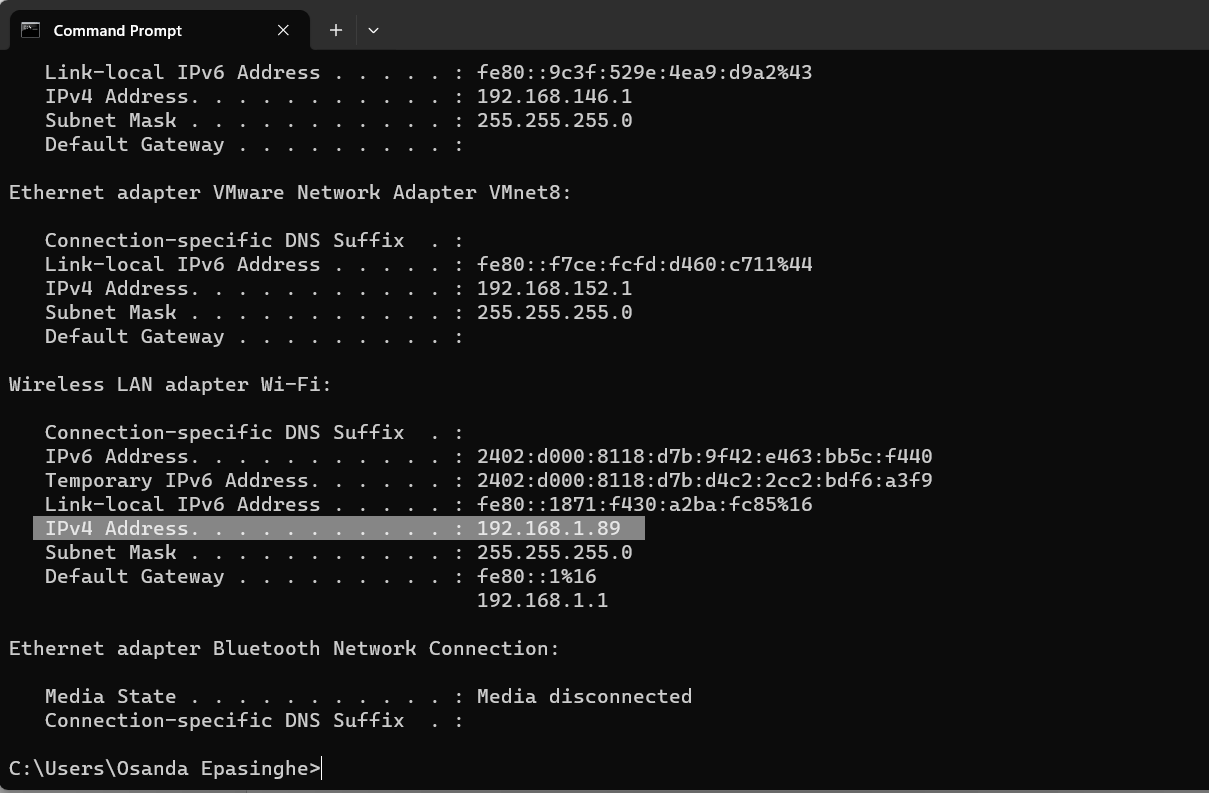
* If you want to separate the files in to two or three machines (for three tier system), please do so in a preferred way.
  + **Some methods:** 
    - If using a Virtual Machine, you can copy paste either tier to the VM
    - Else share it via a USB or Email.

Once that setup is completed, please refer before on how to run the two systems.

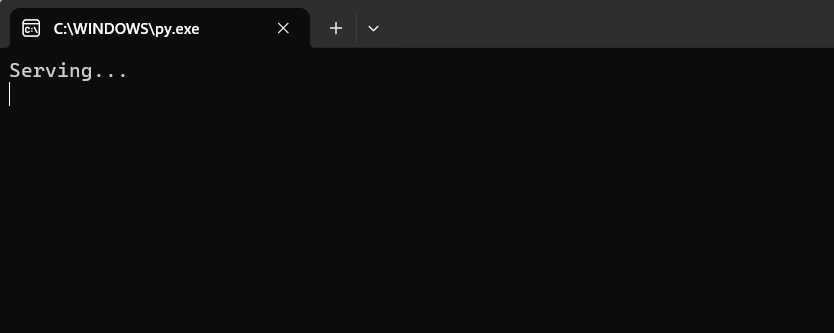
**Two Tier:**

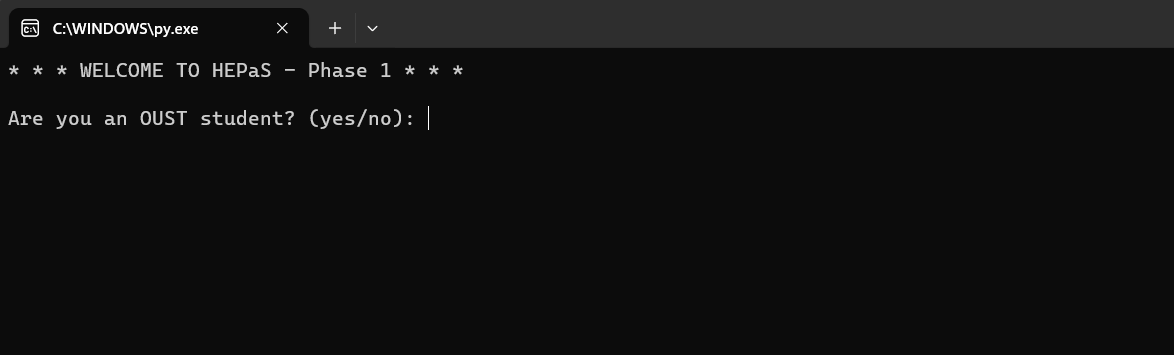
* Firstly, open the phase1client.py using a text editor or a preferred IDE.
* Locate, the line of code in the main function as shown below and replace the IP address, to the IP address of the server machine.



* If you do not know how to find your IP address. Open CMD (Start > type CMD).
* In CMD, type “ipconfig” and locate your network, and copy the IPv4 address.  
  
* After entering the IP address in the phase1client.py , save the file and close it.

**Running the system:**

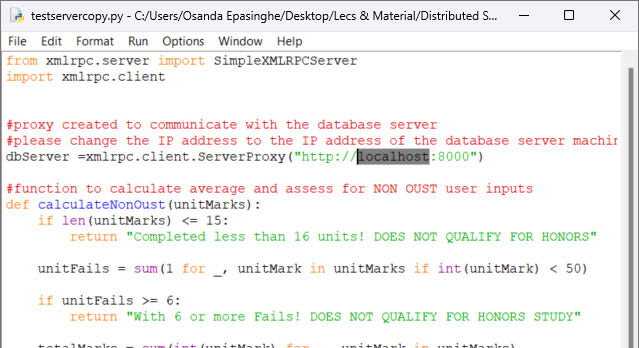
* Next on your server machine, double-click on phase1server.py to open it.
* It should open a window as shown below:
* This means the server is up and running. (Do not close this, it will sever the connection with the server and the client wont work)
* Once the server is initiated, go to your client machine and open phase1client.py   
  This shall open the client UI that enables the client to interact.



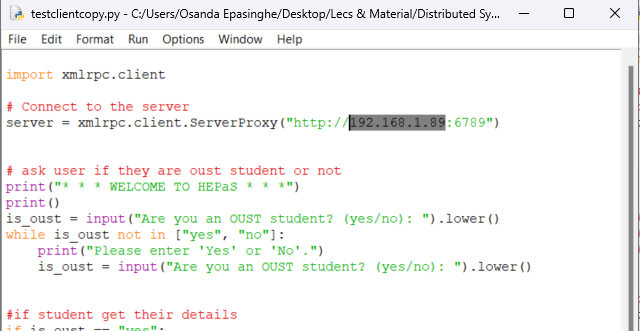
* Enter the required information accordingly and you will be provided with an output.

**Three Tier**

* After separating the files (client, server, and database) into separate machines.
* Obtain the IP address of the machine that holds the database server.
* Next open the server.py on an text editor or IDE on the server machine.
* Locate the line of code as shown below, and replace the highlighted part with the IP address of the database server.



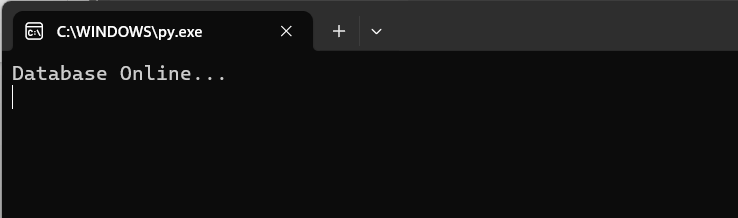
* Save the server.py file and close it.
* Next, obtain the IP address of the machine that holds the server.
* Open client.py on a text editor or IDE on the client machine, then locate the line of code as shown below, and replace the highlighted part with the IP address of the server machine.



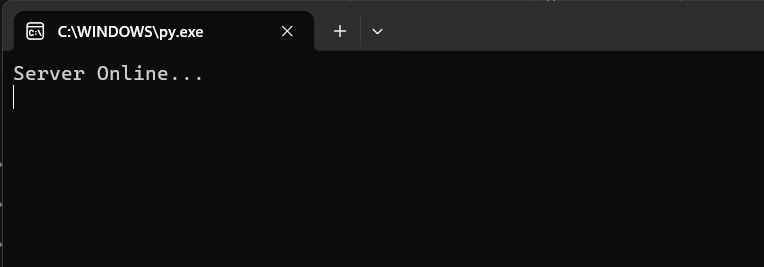
* Save the client.py file and close it.
* Once completed, lets run the system.

**Running the system:**

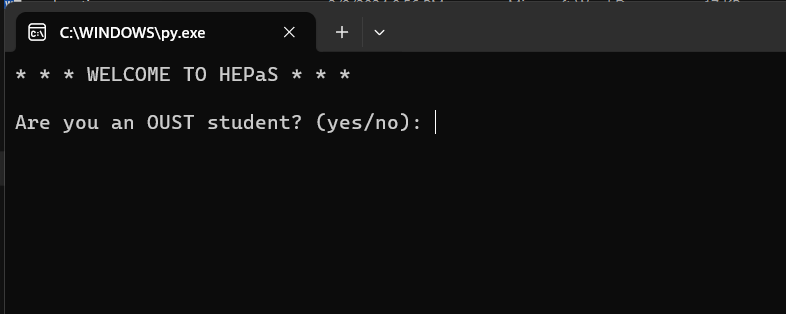
* Double- click and open the dbserver.py on the machine where the database server machine is located.



* A window will open and show the above output
* This should initiate and run the database server.
* Minimize and keep it open, do not close the window as the rest of the steps wont work without it.
* Next, on the machine with the main server, double-click and open server.py .



* A window will open and show the above output
* This should initiate and run the main server.
* Minimize and keep it open, do not close the window as the rest of the steps wont work without it.
* Next, on the client machine, double- click to open the client.py file.
* This will open the Client UI for the three tier system that will allow the users to interact with the system.



* Input the required details accordingly and you will get an output.

**Note: If you cannot open the files by double-clicking, please open them through your IDE and run them.**

**Test Cases**

**Test Case 1: Non OUST Student**

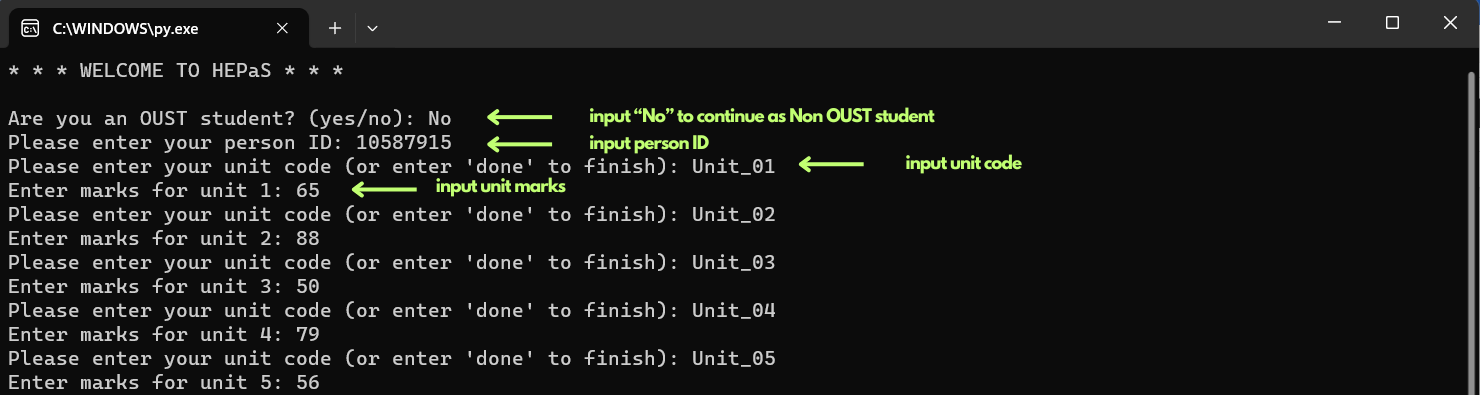
**Input:**

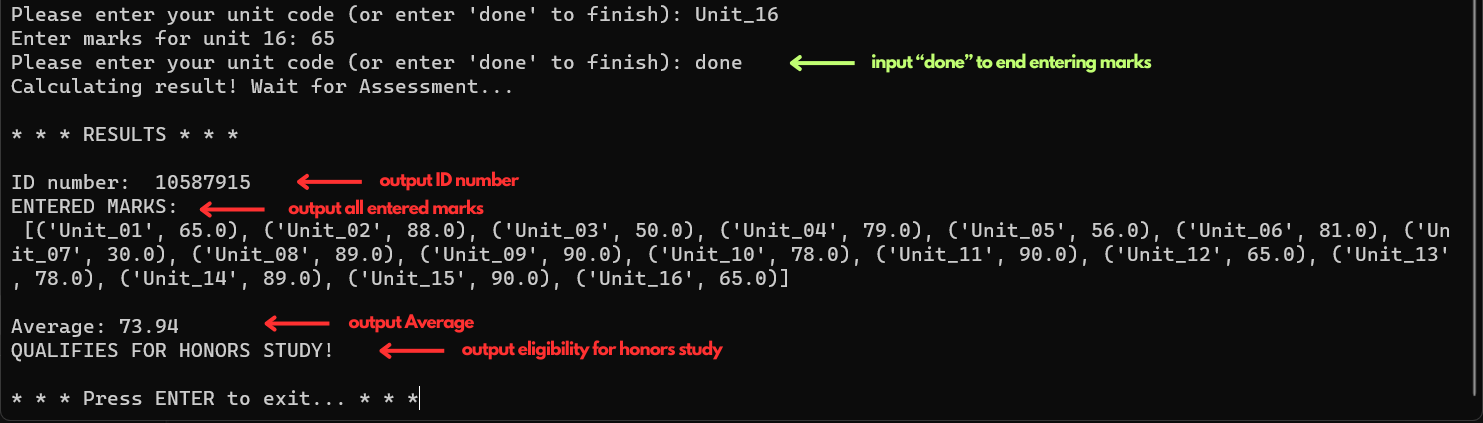
* The user selects the option to indicate they are a non OUST student by entering “No” in the interface when asked “Are you an OUST student? (yes/no): “
* After selecting, the client requires to enter their person ID, and a sequence data containing the unit code and unit marks for the Honors assessment.
* The user submits their Person ID and unit related data by entering “done” in the interface

**Output:**

* The server calculates the average based on the provided data and generates an assessment to indicate of their eligibility of honors study.

**Screenshot examples**

****

****

**Test Case 2: OUST student**

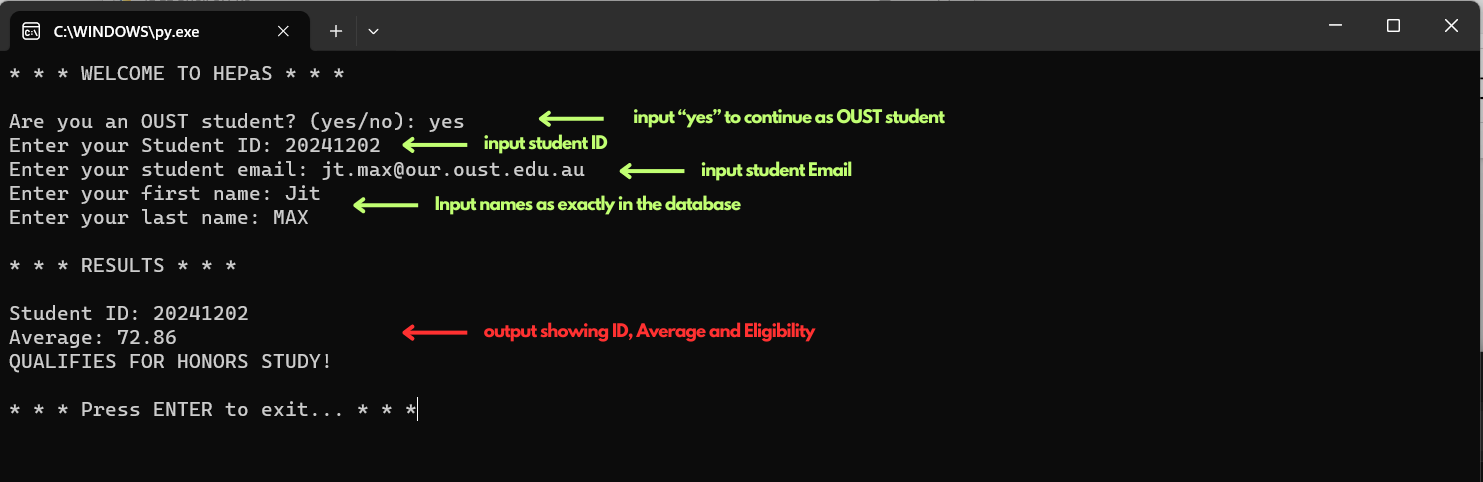
**Input:**

* The user selects the option to indicate they are an OUST student by entering “Yes” in the interface when asked “Are you an OUST student? (yes/no): “
* After selecting, the client requires to enter their Student ID, Student Email, Student first name, Student last name. (These should be entered exactly as provided to the database)
* The user submits their details by pressing enter.

**Output:**

* Server calculates the average based on the retrieved data from the database and generates an assessment to indicate of their eligibility of honors study.

**Screenshot examples**

****

**Test Case 3: Unauthenticated User**

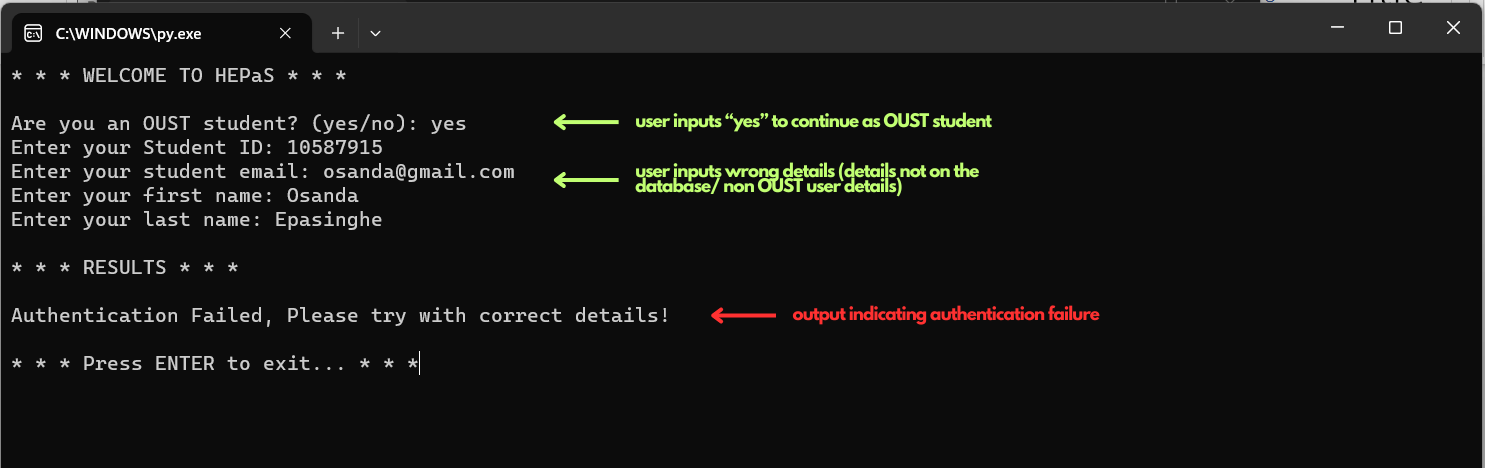
**Input:**

* The user selects the option to indicate they are an OUST student by entering “Yes” in the interface when asked “Are you an OUST student? (yes/no): “
* Client requires to enter Student ID, Email, First name and Last name
* User inputs incorrect details
* User submits their details by pressing enter

**Output:**

* Server checks user details against the database, and generates a message indicating authentication failure. (When authentication is failed, no access to database is granted)

**Screenshot example:**

****

**Test Case 4: Non OUST user entering less than required marks**

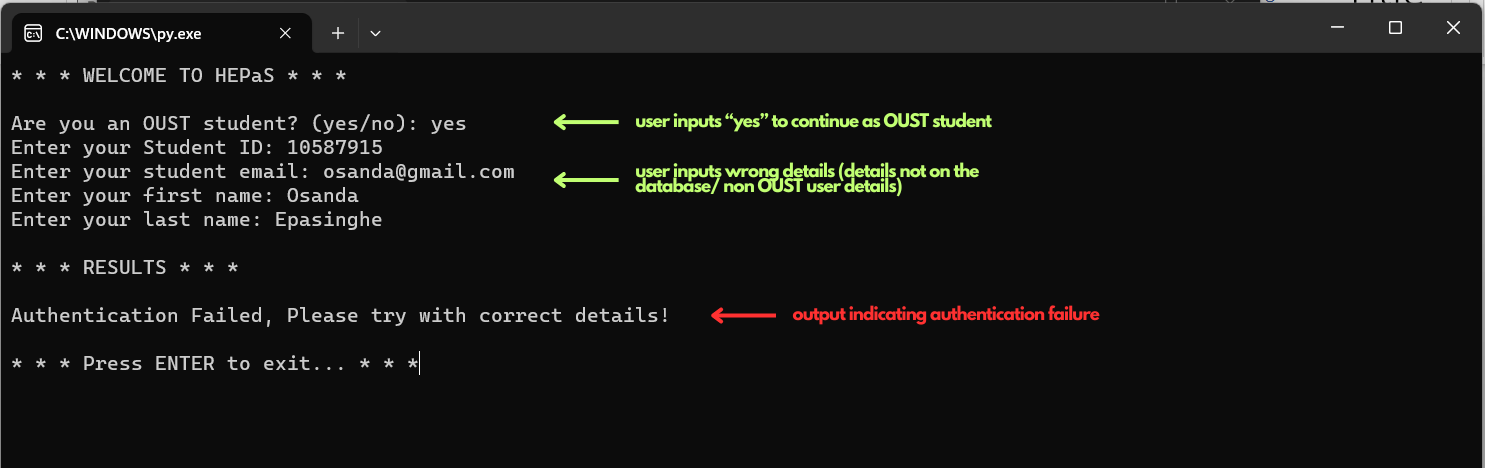
**Input:**

* The user selects the option to indicate they are a non OUST student by entering “No” in the interface when asked “Are you an OUST student? (yes/no): “
* After selecting, the client requires to enter their person ID, and a sequence data containing the unit code and unit marks for the Honors assessment.
* The user submits their Person ID
* The user enters less than 16 unit related data and submits by entering “done” in the interface (either the user has only done less than 16 units, or has forgotten to enter all before submitting)

**Output:**

* The server checks the requirement of at least 16 units completed.
* The server finds less than the minimum requirement is only provided and generates an assessment indicating less than 16 units are completed which does not qualify for honors study

**Screenshot example:**

****

**Test Case 5: Improper inputs (Student & Person ID, Marks)**

**For student & person IDs**

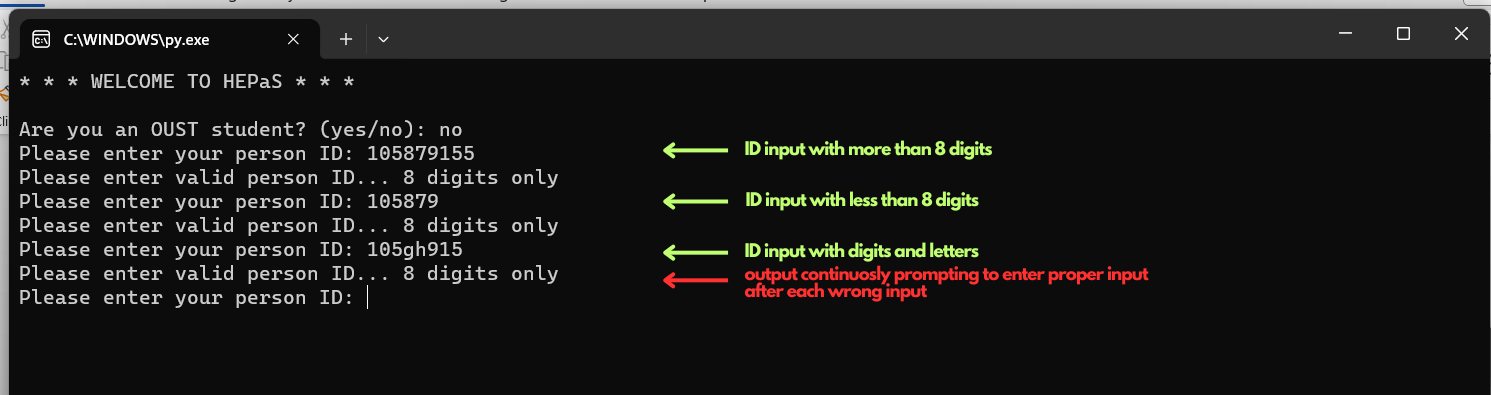
**Input:**

* User inputs ID numbers not consisting of 8 digits or consisting different characters

**Output:**

* System keeps prompting error handling message to correct the input. In this case, it asks to enter only 8 digits.

**Example screenshot**:



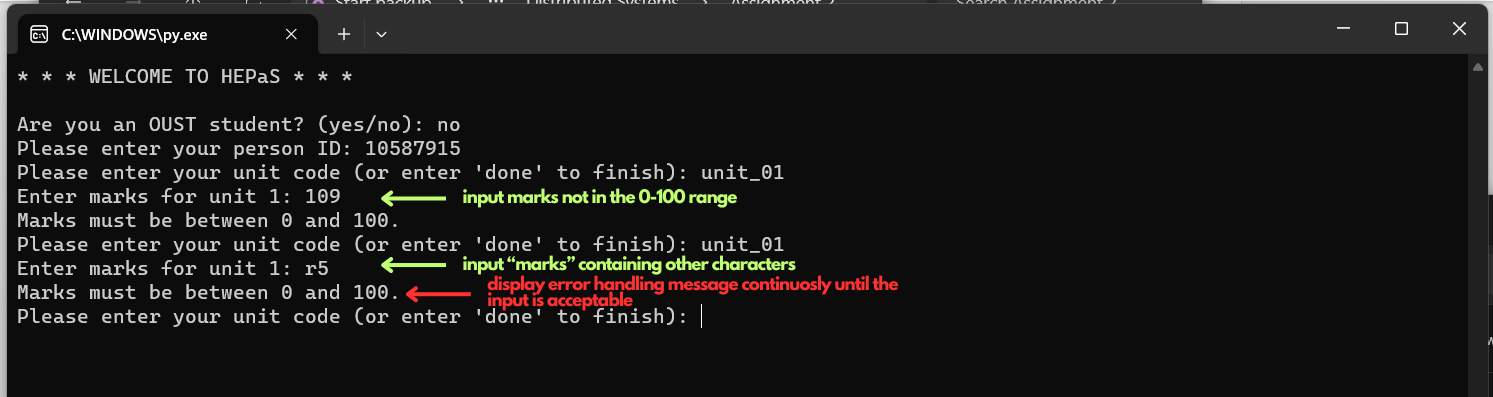
**For marks**

**Input:**

* User inputs marks not in range of 0-100
* User input “marks” containing other characters

**Output**

* System keeps prompting error handling message to correct the input. In this case, it asks us to enter marks in the acceptable range

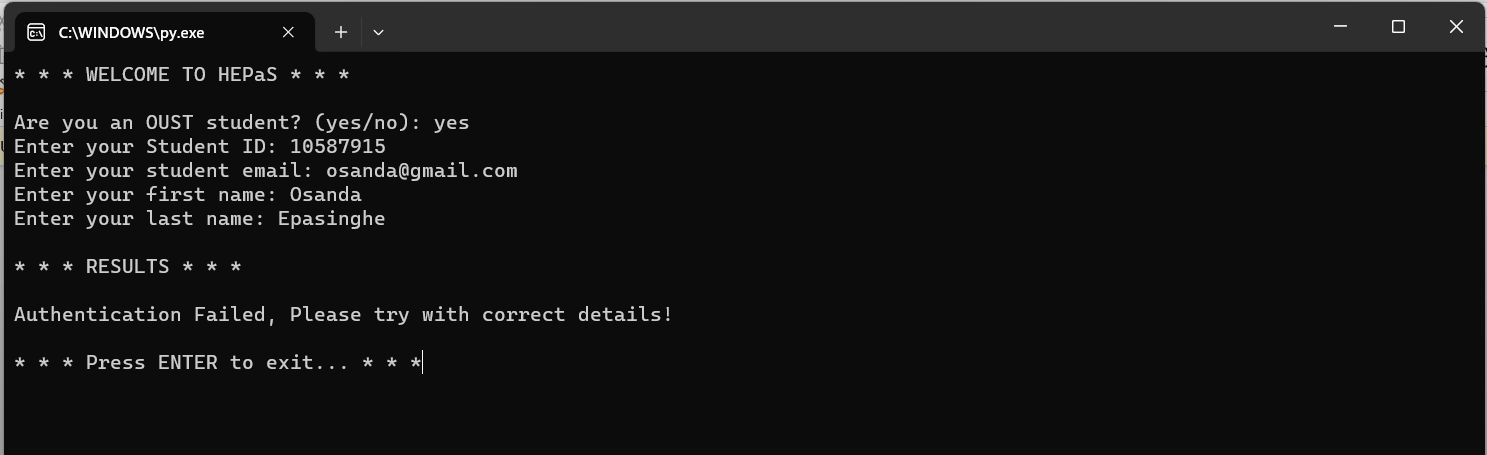
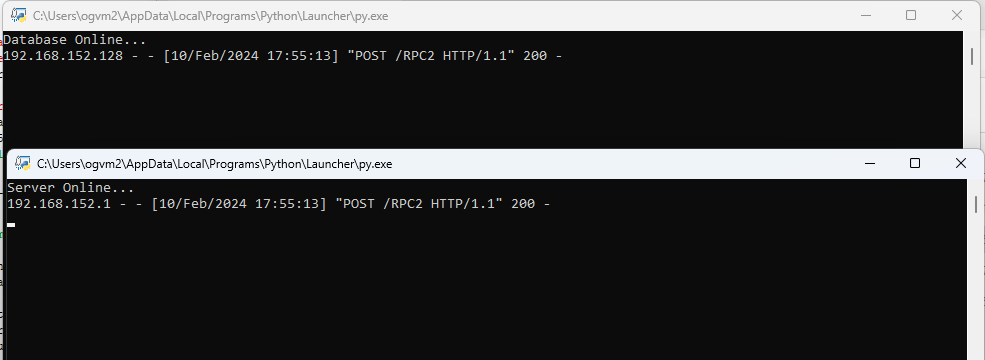


**Screenshots of actual run on different machines (VM and User PC)**

* In this setup, the VM held the database server and the main server.
* The IP addresses were replaced according for the situation.

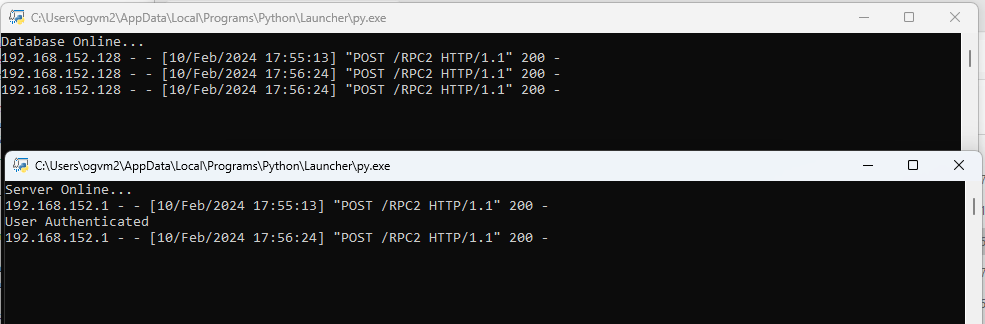
**Screenshots:**

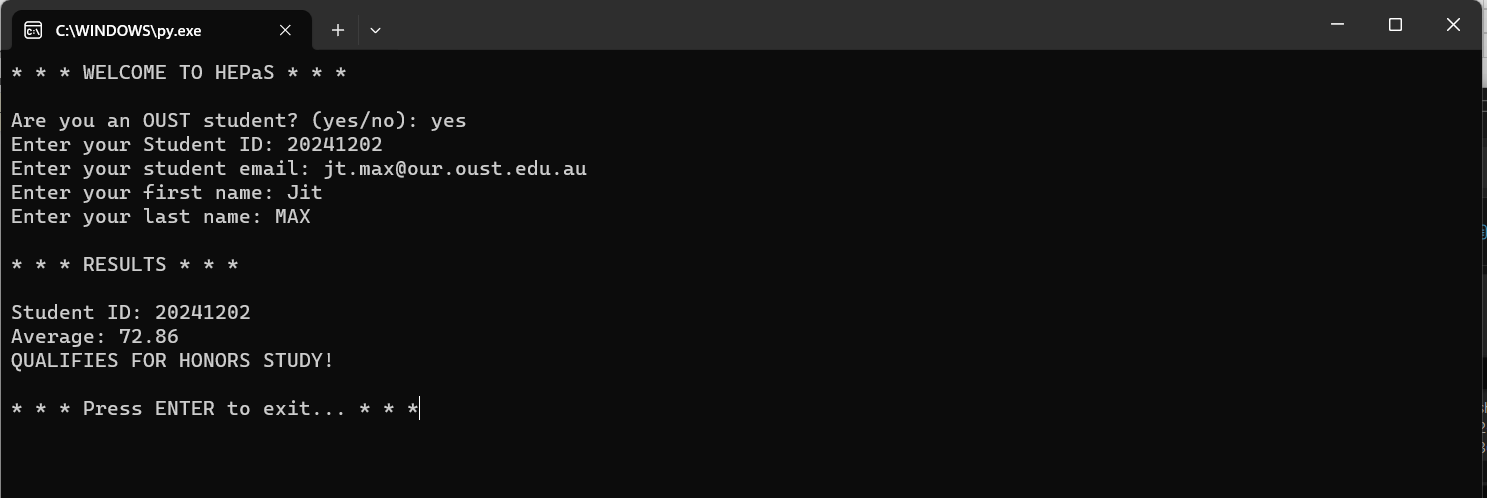
1. **Using the system as an OUST student – Inputting incorrect details**



When incorrect details are input into the system, it will display a message indicating authentication failure.   
It’s important to notice that from the server and database record logs provided in the screen shots that, when the details are input and submitted, the server accesses the database to authenticate user, but since the user details are incorrect the server doesn’t find relevant information in the database, therefore returns an authentication failure message.

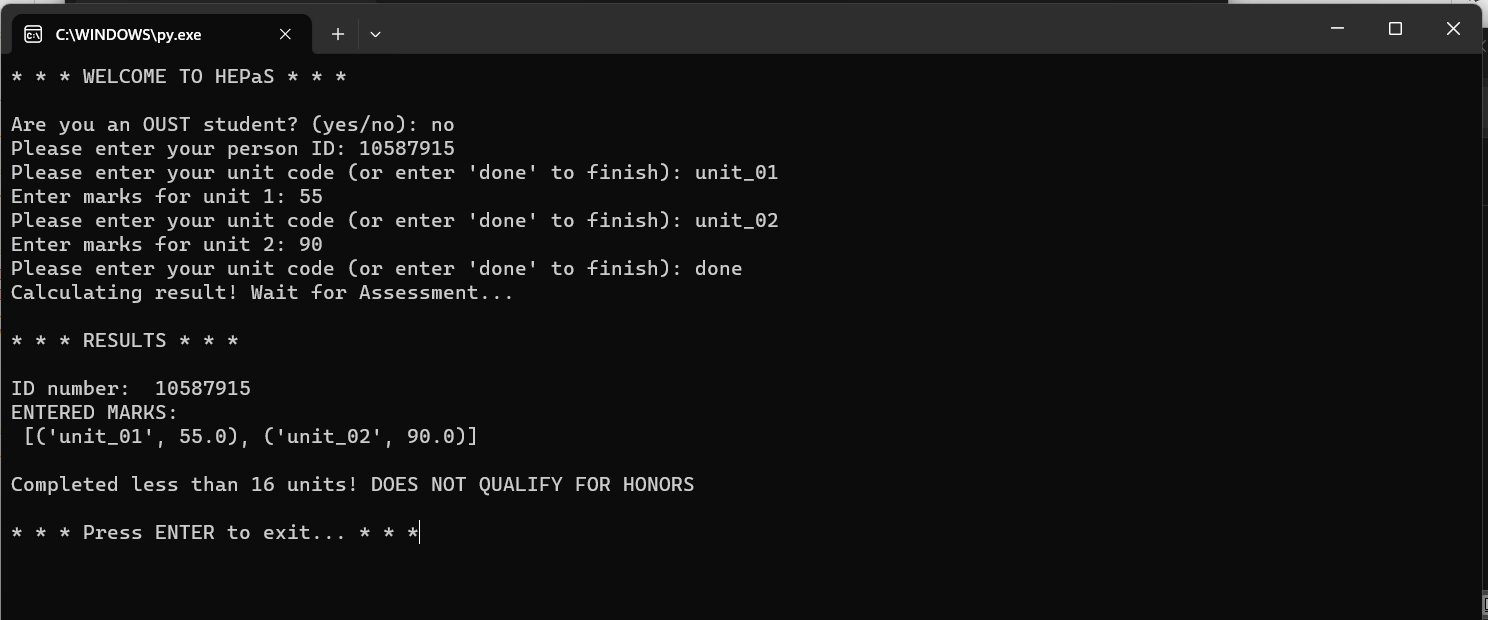
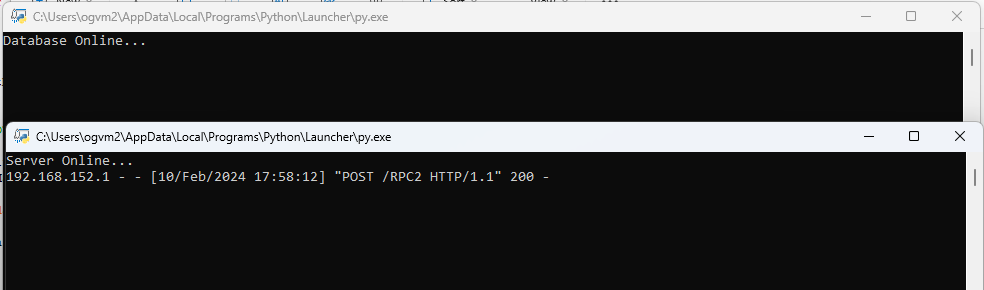
1. **Using the system as an OUST student – Inputting correct details**





When correct details are input into the system, it will display the average and eligibility for honors study.   
In this encounter, according to the record logs, the server accesses the database to check the provided details. Once the server authenticates the user against the database details, the server gets access to retrieve the other course related data from the database.

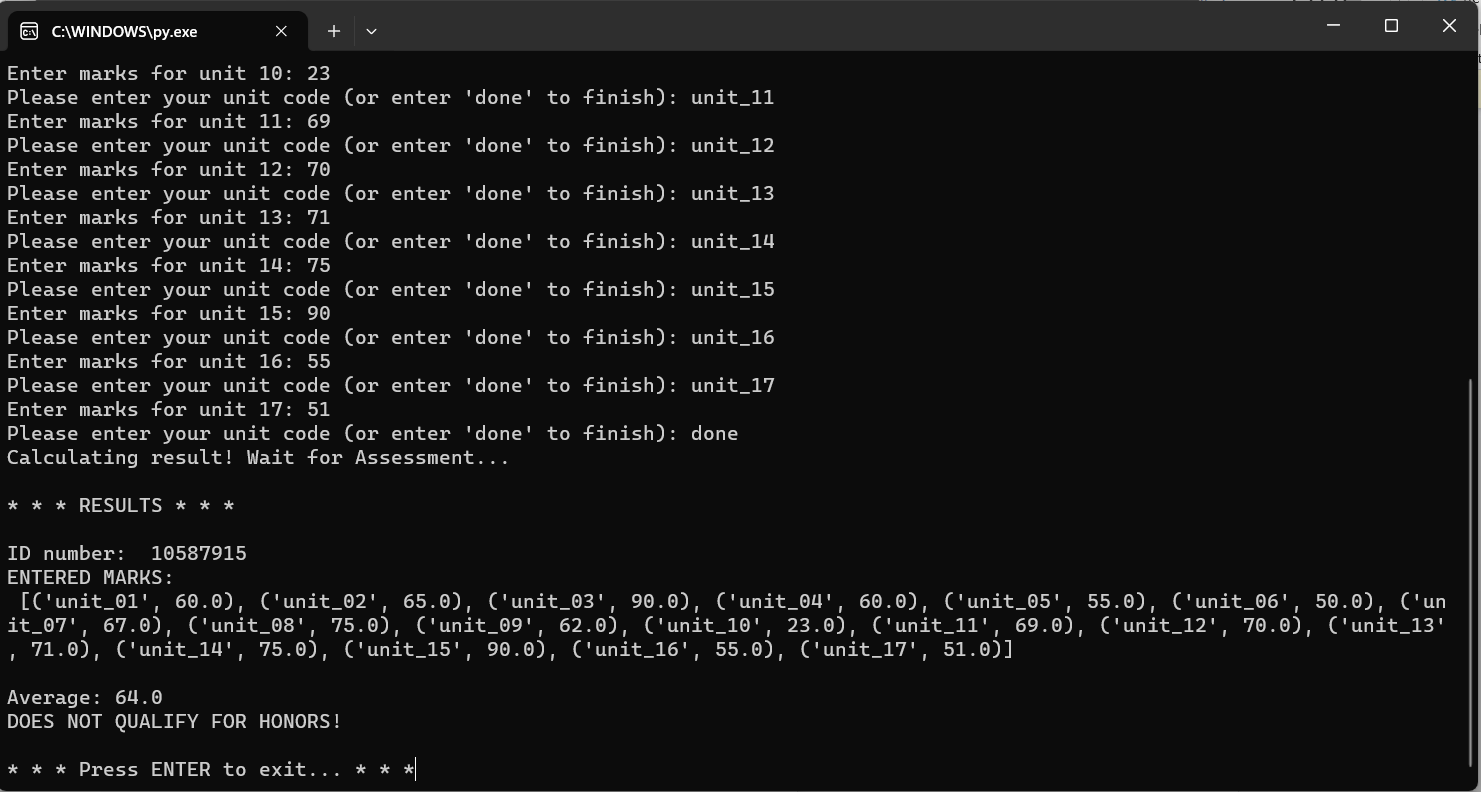
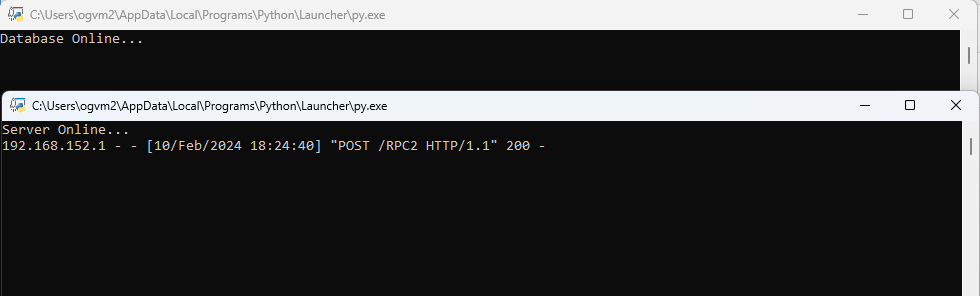
1. **Using the system as a non OUST student (entering less than required data)**



In this, the user enters less than the required amount of unit marks and gets an output mentioning the ineligibility for the honors study.

In this encounter, we can clearly see that the database is not accessed (because they are non OUST student) and that only the server does the processing.

1. **Using the system as a non OUST student (entering appropriate amount of data)**



In this, the user enters the appropriate amount of unit marks and gets an honors study eligibility assessment output according to the average.