Benny Lee - GoSchool - Assignment 5 - Write Up

**The components implemented in this assignment**

The files consist of the console application and the REST API web service.

The console application allows the user to perform CRUD of courses information using the REST API service and simultaneously storing the information in MySQL, viewable via the software MySQL Workbench.

**Specifically, the client application allows user to:**

o Add course (1 course at a time, receiving input for course ID and course title)

o Update course (1 course at a time, receiving input for course ID, and only can update the course title)

o Delete course (1 course at a time, receiving input for course ID)

o Retrieve course (Can view all courses or manually check 1 course at a time via browser/curl)

**The REST API allows courses to be:**

o Created

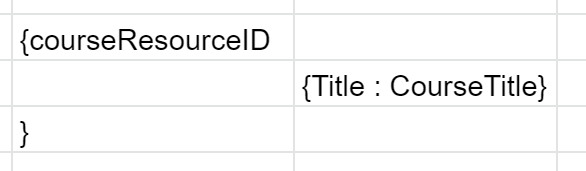
o Updated

o Deleted

o Retrieved   
  
**All course information is stored in a MySQL database.**

This happens together when information received from user, is passed to the REST API service.

The courses JSON structure will be as follow :

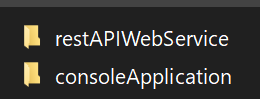


The courseResourceID will be unique, as it is to be stored in the primary key column in MySQL, and it is also unique to be the resource in the REST API URL.

The title is there, so that this courseResourceID can be expanded to include more information with other key value pairs. For the purpose of this assignment, it was set to only include the “Title” as key and value as “CourseTitle”.

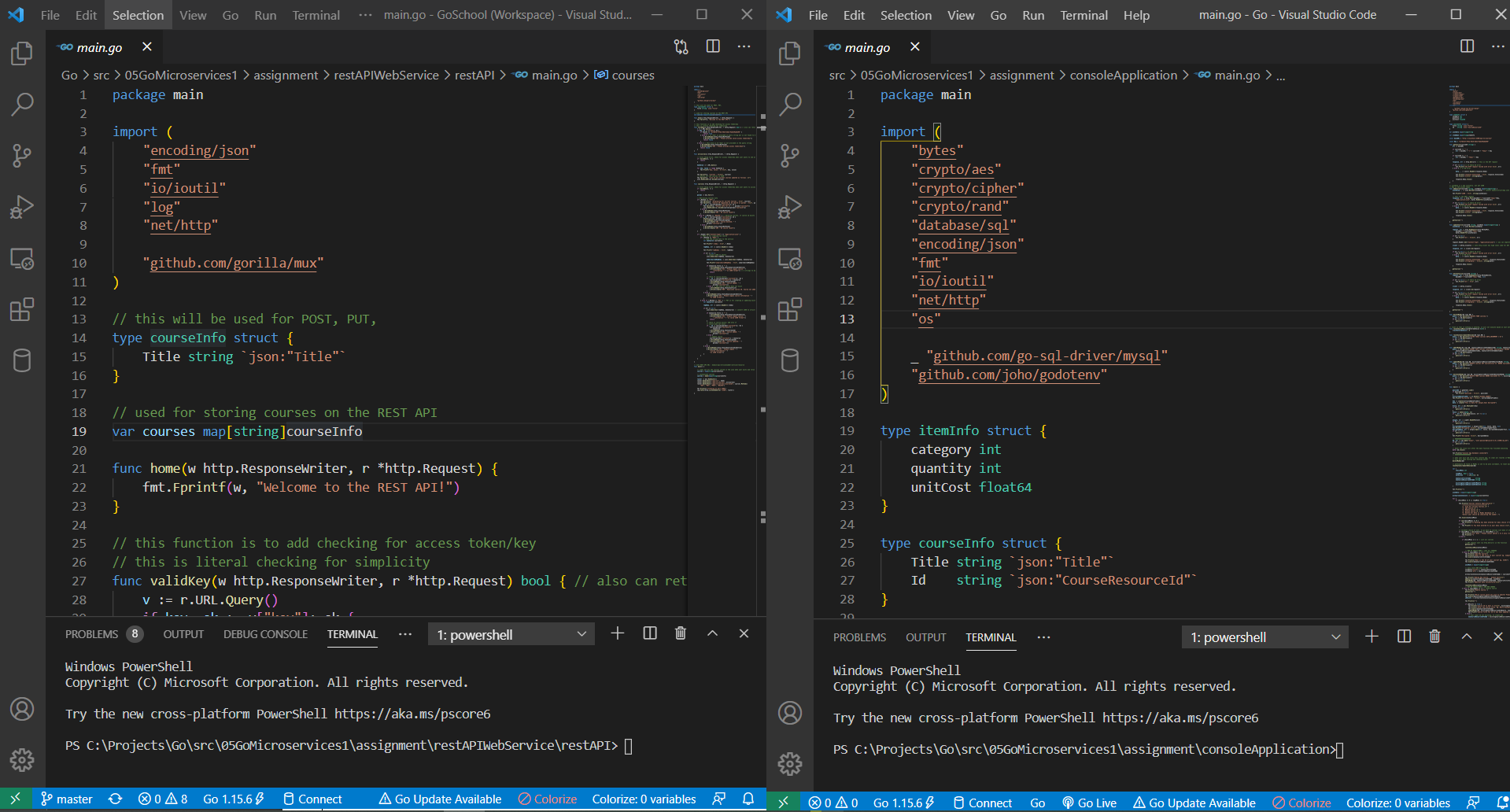
**A setup guide on how to deploy the application**

As I am using Windows OS and Visual Studio Code, all my screenshots and instructions will be for Windows 10 and VSC only.



There are 2 folders as shown above and inside are 1 main.go file in each folder.

Both main.go files are to be opened and run at the same time, preferably in different IDE windows for ease of viewing.



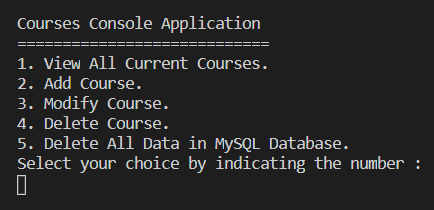
Navigate to their respective folders in the Terminal :

REST API : cd C:\Projects\Go\src\05GoMicroservices1\assignment\restAPIWebService\restAPI

Console Application : cd C:\Projects\Go\src\05GoMicroservices1\assignment\consoleApplication

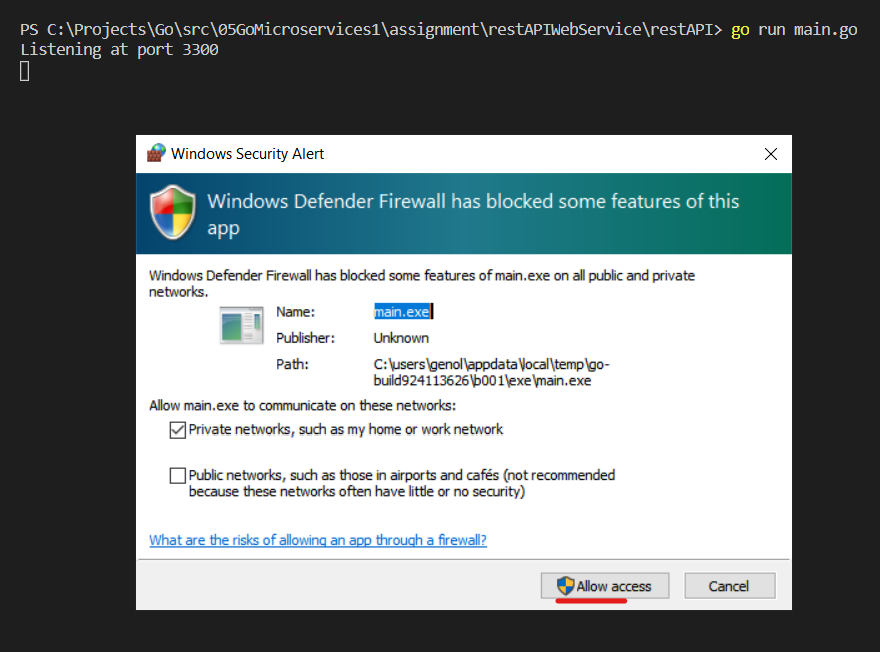
Type in “go run main.go” without quotes at both terminals to start both applications.

The REST API will be handling the requests from the Console Application as the user make their choices as presented in the list of options :



This is what is to be seen in the Console Application.

As for the REST API, if you are using Windows, you are likely to receive a Windows Security Alert. Just click on “Allow access” to continue

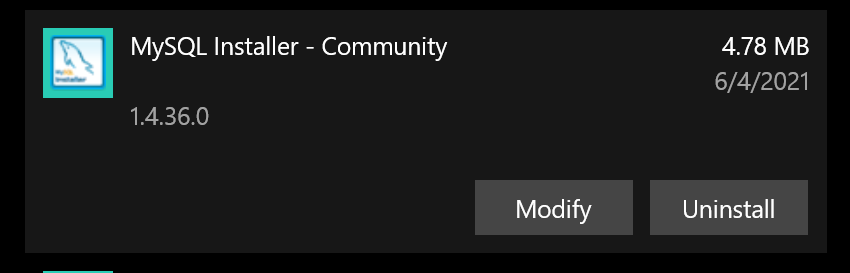


Next, you will need to download and have MySQL Workbench installed

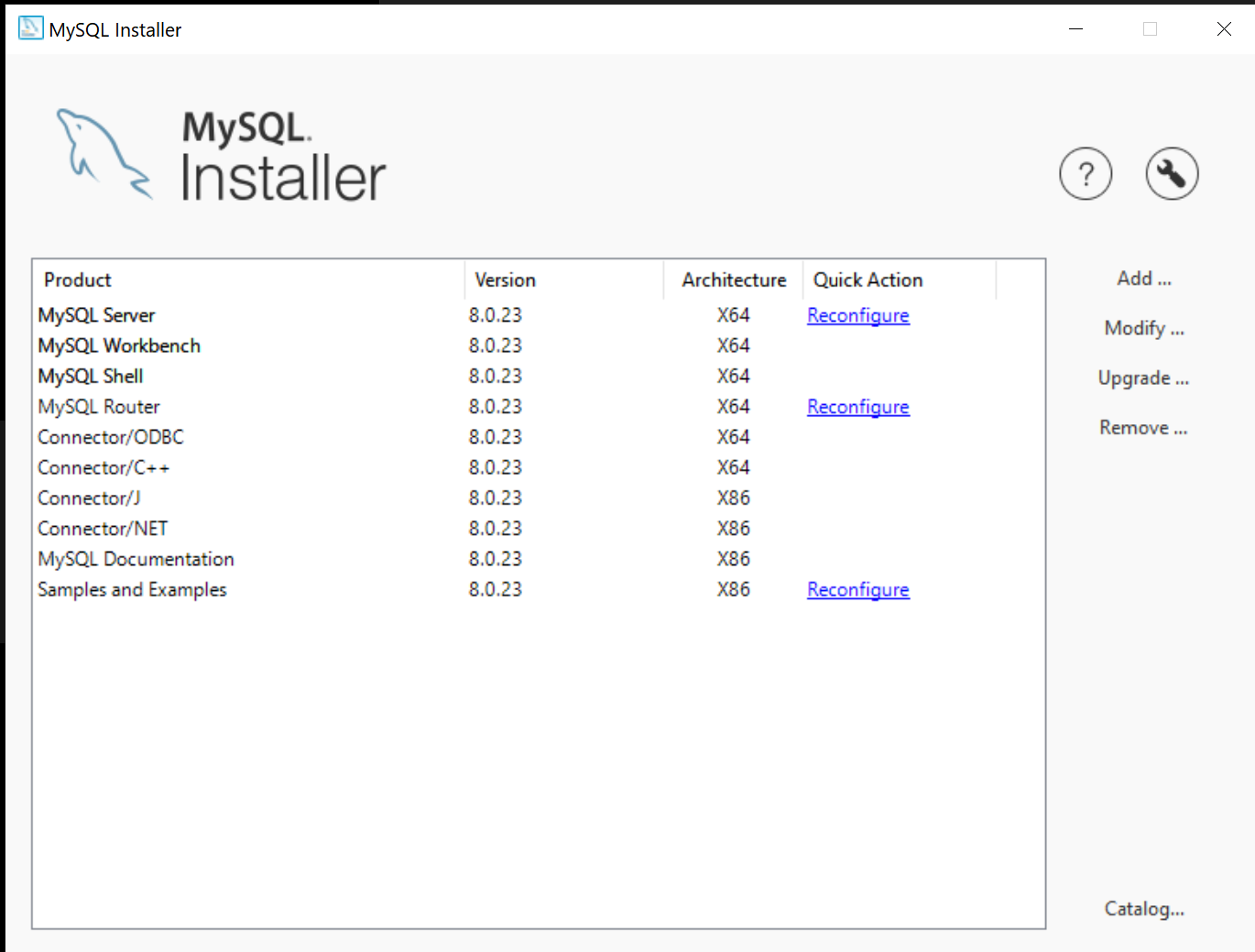
<https://dev.mysql.com/downloads/workbench/>

I have selected to install all components

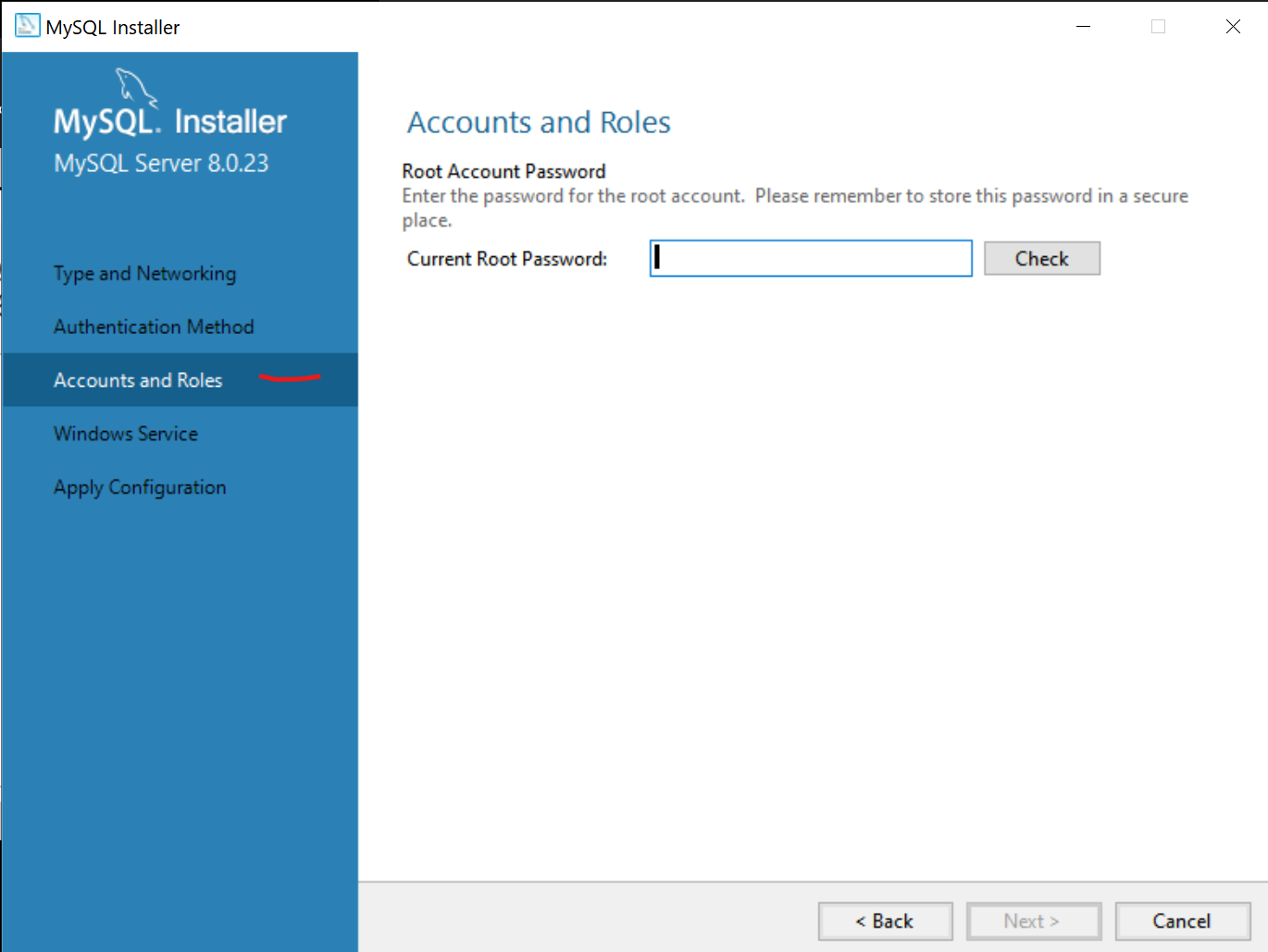
After installation,



In the Add or remove programs under window settings, find the above program and click on “Modify” to open this window



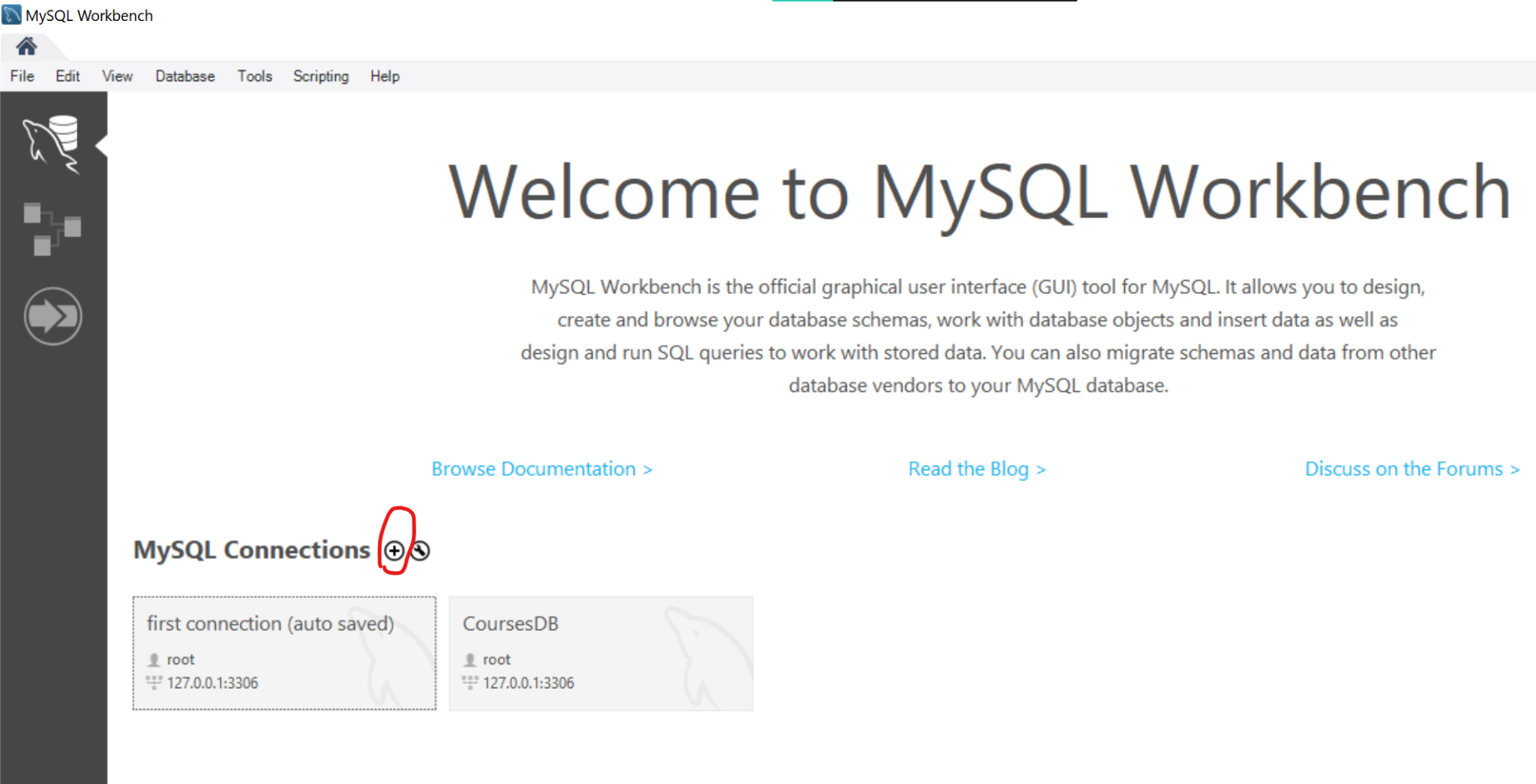
These are the components I have installed. Click on Reconfigure under MySQL Server to open up this window



Go through the 5 stages with default options, but especially under “Accounts and Roles”, there is a need to create a user account, password and set role for this user and/or for root user as well. These will be needed each time to access MySQL Workbench to view the database.

If there are unexpected errors, try to disable any antivirus real time protection, and open VSC and MySQL Workbench as administrator.

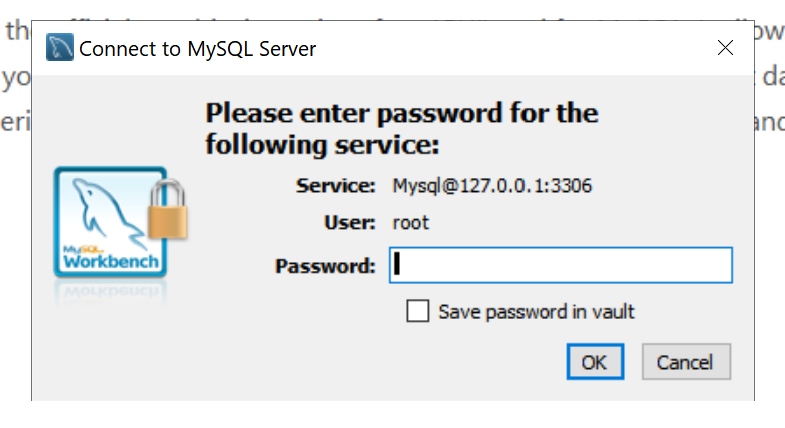
Open up MySQL Workbench and you should see a grey connection card, if not click the plus sign



Configure the connection port to localhost:3306

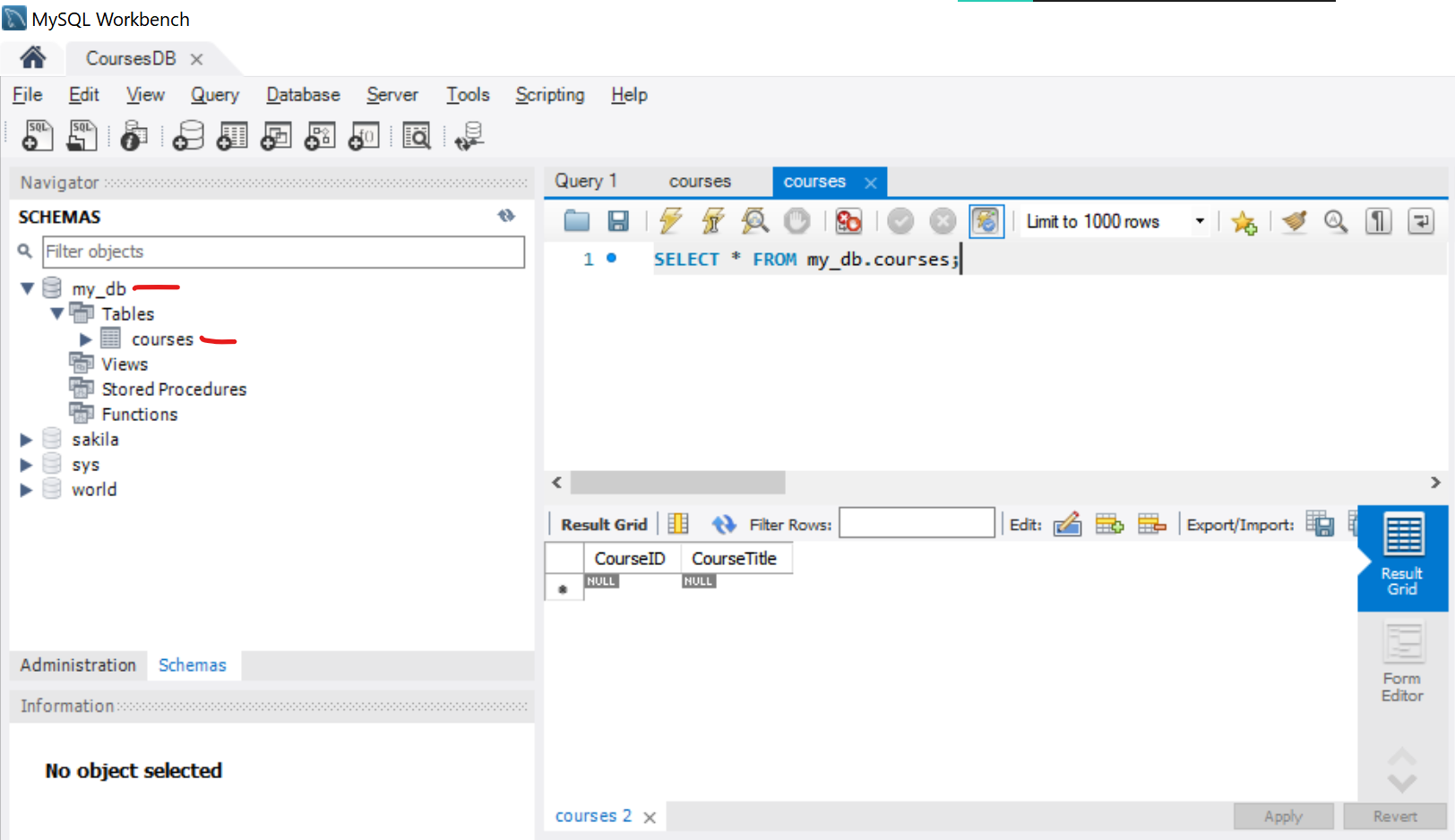
I have named mine as CoursesDB.

If you have setup your password, it will then request for your password

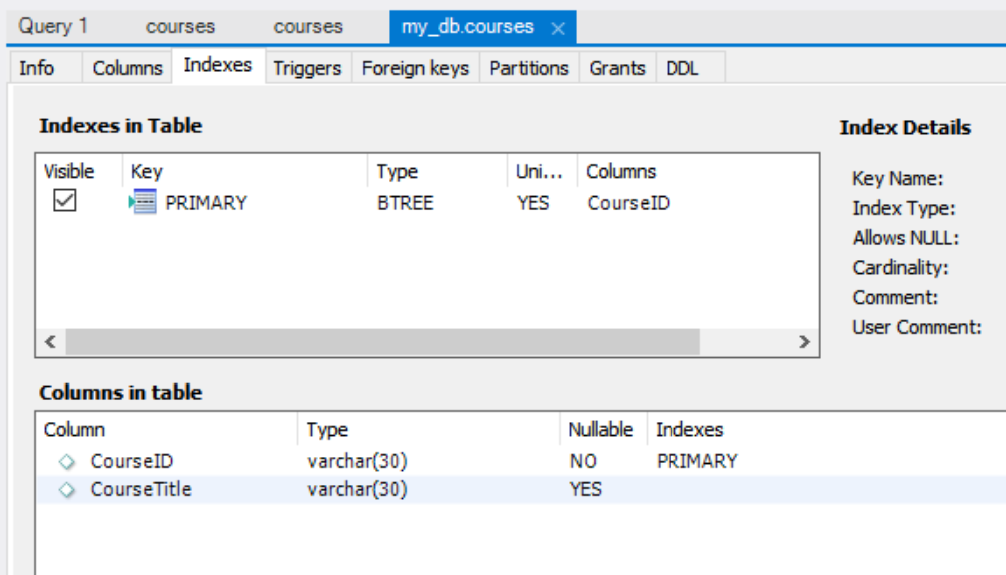


When you go in, create a database. I called mine my\_db as can be seen here.

My table name is called “courses”



The courses table i created 2 columns



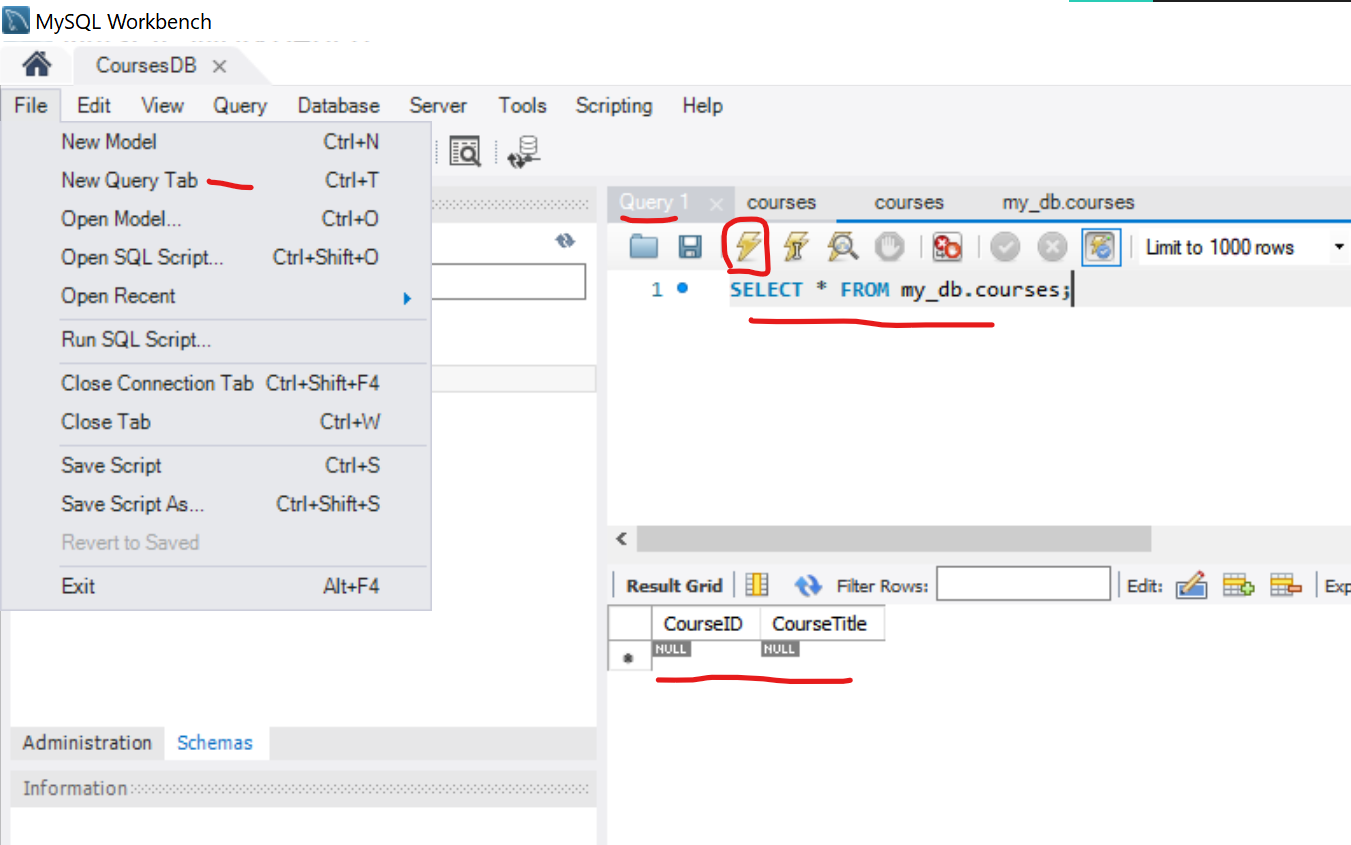
named as CourseID and CourseTitle. These are the specific options I gave them.

CourseID is type varchar(30), the primary key and is not nullable.

CourseTitle is type varchar(30), and is nullable.

When this is done, you can open a new query tab to key in MySQL statements like

Select \* from courses to see all the info from the table courses



When you press the Lightning button to execute the MySQL query, another windows below would pop up to show that the courses table is empty.

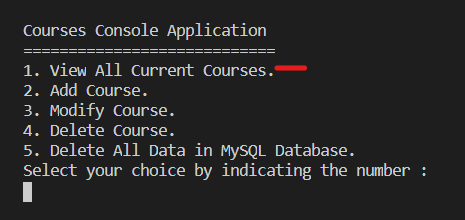
The setup is now completed, we can now proceed to test the application.

**A guide on how to test the application**

Some remarks to note : Should there be any errors faced in the course of testing, please close both applications and restart them again. This is to reset the data in both sides. If only 1 of the app is closed and restarted, some info may be retained that becomes uneditable.

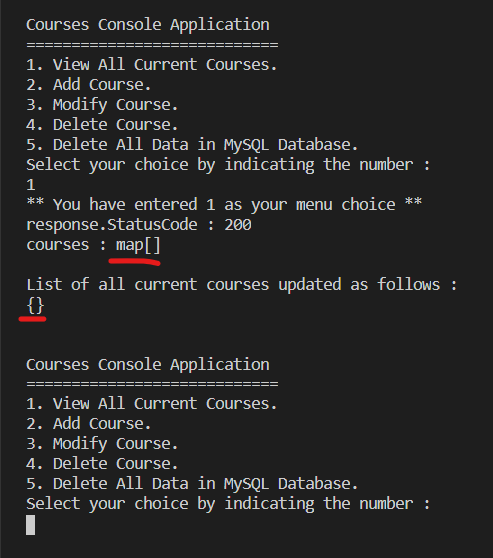
There is not a lot of validation. Assumption here is each Course ID input keyed in by user for creating new, is always unique, and that user input is always accurate and exact. Any deviation from the guide may cause unexpected behaviour, errors and closing of applications.

To start, in the Console Application, key in 1 and enter



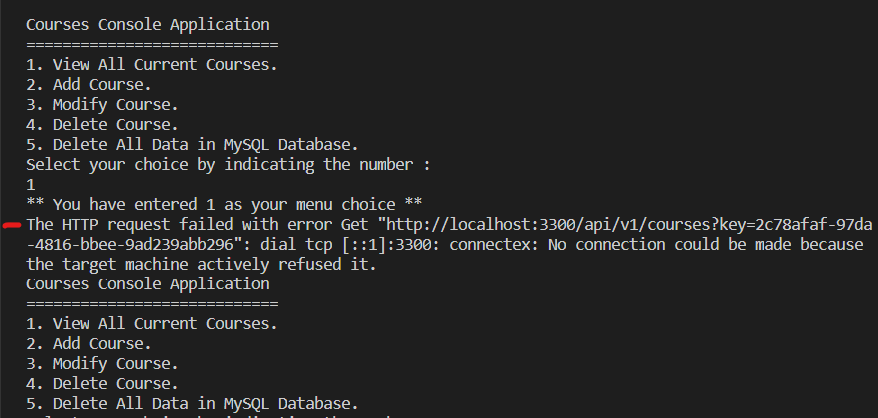
**Retrieve** : You would see an empty map and empty JSON because we have not input anything yet.

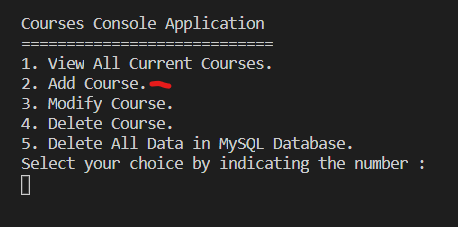
The map is stored in the client side while the JSON is stored in the REST API side.



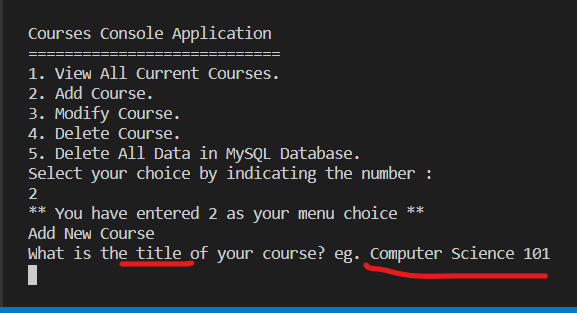
The console application will always then loop back to this main menu for user input again.

*Note : If the REST API is not running. You will get an error in the console application instead of the empty map and JSON. So make sure the REST API runs first before any user input.*



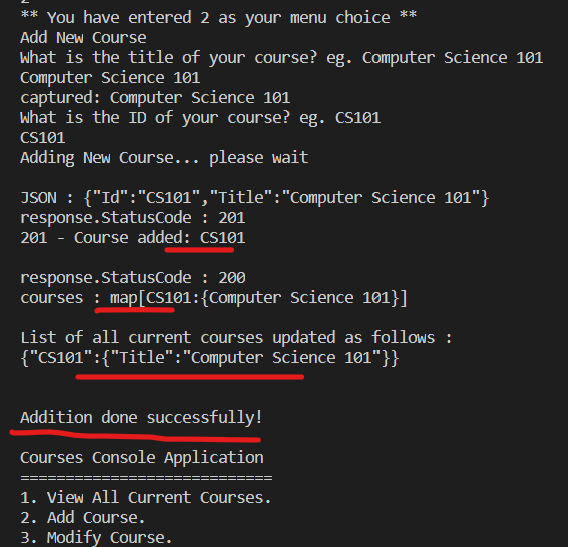


**Create / Add** : Key in 2 and enter



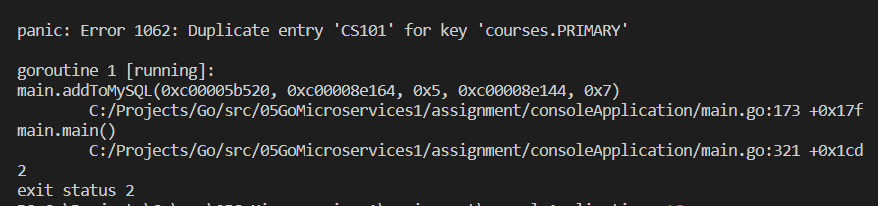
Here you can enter a new course. Key in the title first, up to 30 characters like “Computer Science 101”

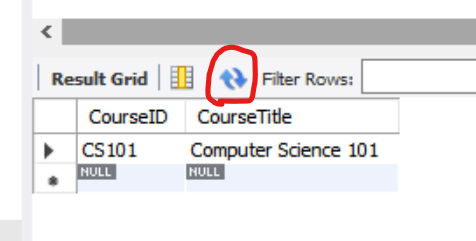
It will then ask for how you want to name your Course ID, up to 30 characters like “CS101”



When you are done with these 2 inputs, you should see that the Course ID and Title has been added into the MAP and JSON.

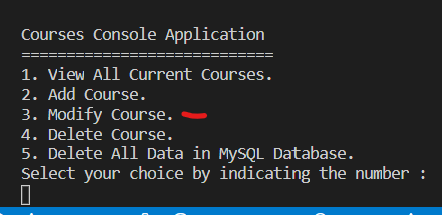
If the Course ID is not unique, it will throw an error and the application will close and both applications will have to be restarted. All info keyed in previously will be removed.



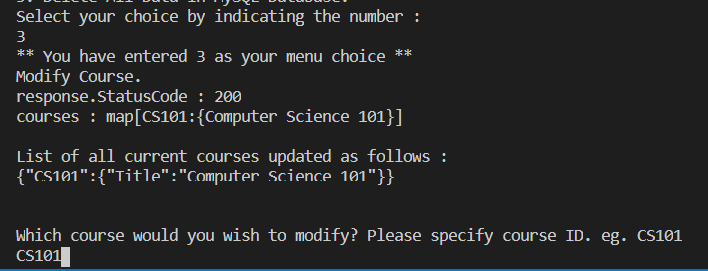


You can go to MySQL Workbench and click refresh button to see that it has been stored in the database.

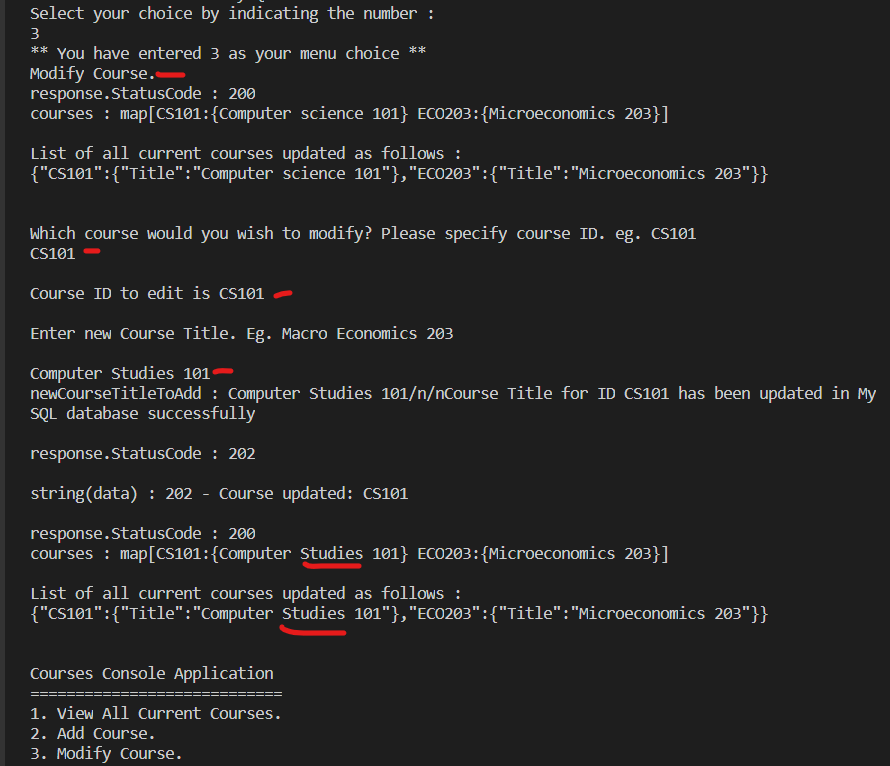
**Modify / Update** : Next we can update an existing entry



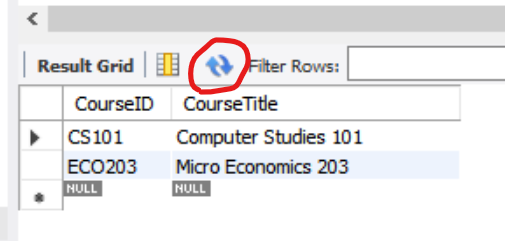
Key in 3 and enter



You get to see the all the current courses and from here you can choose which ID to use.

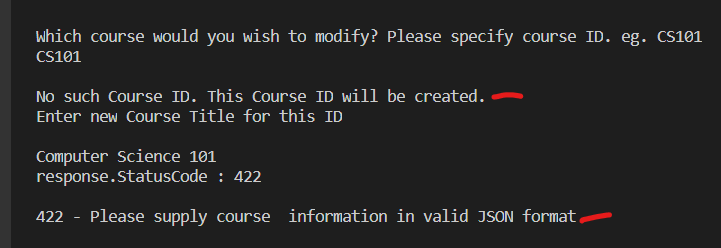


You should see the changes as per input in the MAP and JSON.

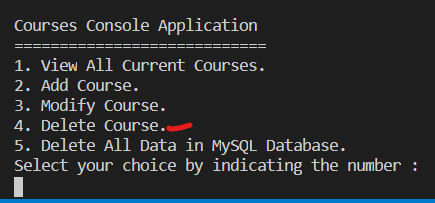


Refresh in MySQL Workbench to see the changes.

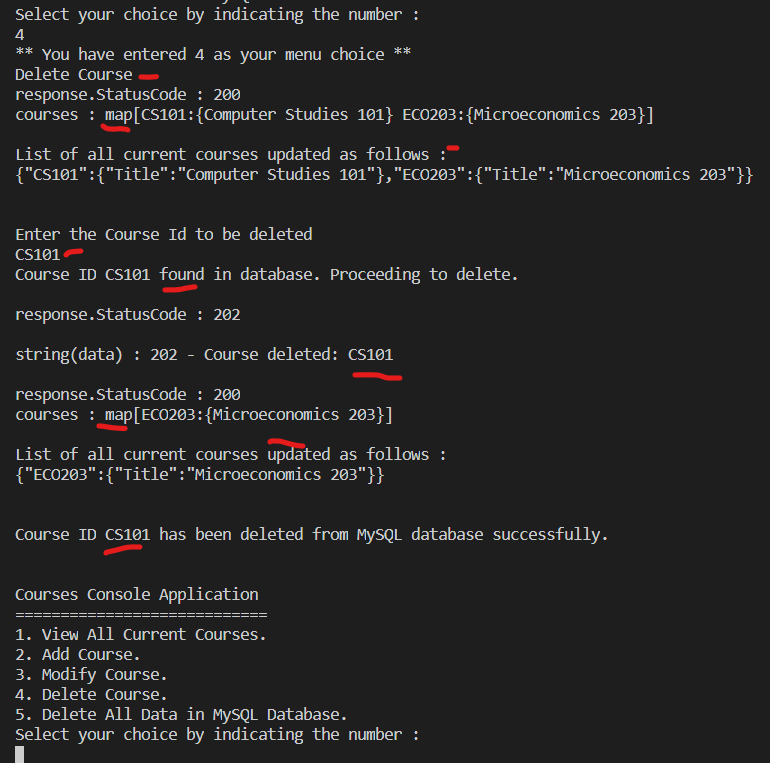
If you choose to Modify / Update, and you key in a Course ID that is not found in the current courses. It will automatically add as new for you and will ask for Course Title.



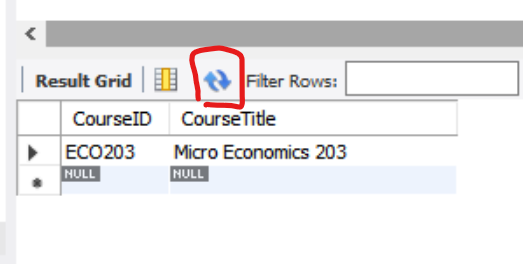
However I have found that while the MySQL Database will be updated, I have not figured out the bug of not adding in the REST API at this time.



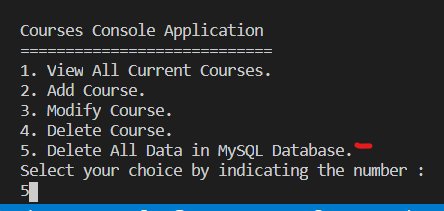
**Delete** : Key in 4 and enter



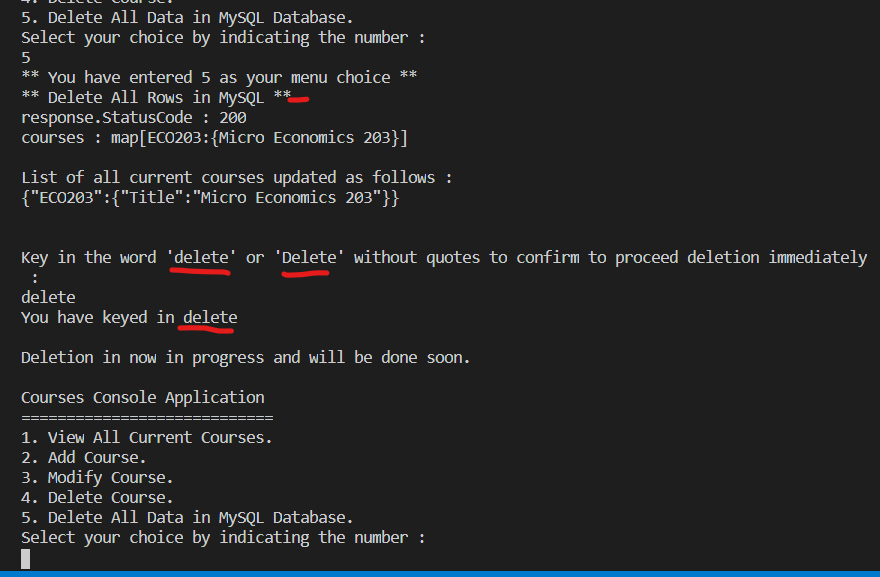
You would see the Course ID keyed in has been found or not and if found, will proceed to delete. And it will be removed from the MAP and the JSON.



Refresh in MySQL Workbench to see the changes.

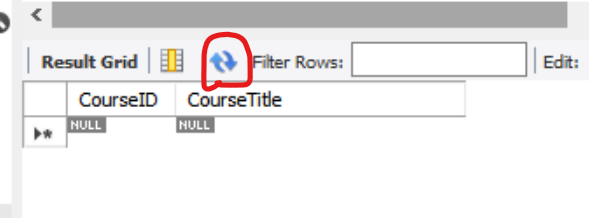


**Delete All** : incase you wish to delete everything in the 2 applications and in the MySQL database to start afresh, key in 5 and enter



Here you will be asked for a last confirmation to proceed by keying in ‘delete’ or ‘Delete’.

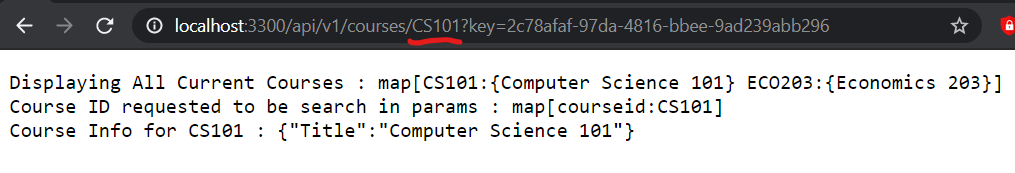
When you proceed to delete, it will be done.

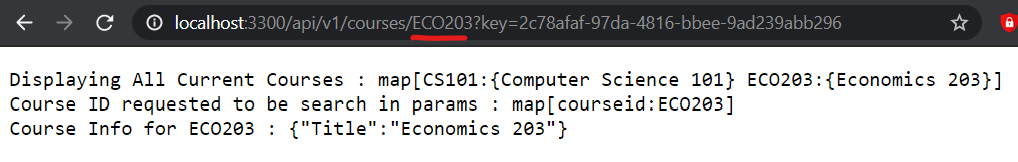


Refresh in MySQL Workbench to see the changes.

However, if you key in 1 to check the MAP and JSON, the data is still there as I have not figured out how to do it via REST API yet at this time. Restart both applications to clear both sides’ data.

Also note that info for individual resource can be access via say a browser. It is not implemented in the console application at this time.





http://localhost:3300/api/v1/courses/**CS101**?key=2c78afaf-97da-4816-bbee-9ad239abb296

http://localhost:3300/api/v1/courses/**ECO203**?key=2c78afaf-97da-4816-bbee-9ad239abb296

Key in the Course ID specifically to obtain all the current Courses, and the info for the specific ID you entered

**Security Keys**

For the security keys, at this time, I have tried JWT, .env file and some others but did not manage to successfully implement fully and have resorted to using URL query string to make it work in view of time.