CIS-344 FINAL REPORT

The final presentation is a product of this report. I will discuss my process and/or progress step by step as I move forward toward my end goal. I will create a database on MySQL, and then fill in the information for tables, primary keys, foreign keys, stored procedures and so on. After that, I will connect MySQL and Python and then use IDLE to run the code. Finally, I will see the result of the code that I ran in visual representation.

So, if you were paying attention in class, the first five steps are simple. It does not require anything other than following the steps and taking your time to get it right. I have images of each step in the appendix of this report. The first step is to create the database. Next, you use the database you created to create tables and relationships within that database. On the other hand, it was necessary to go back and review the lesson on stored procedures to complete the three procedures needed to move on to the next step.

Question six was probably the most mentally taxing for the completion of MySQL information. In order to get the stored procedure portion of the project correct, it is so important to remember the steps. Delimiter, space, double forward slash, create procedure, remembering names, creating names, begin, end, forward slash again, delimiter again, and finally semicolon.

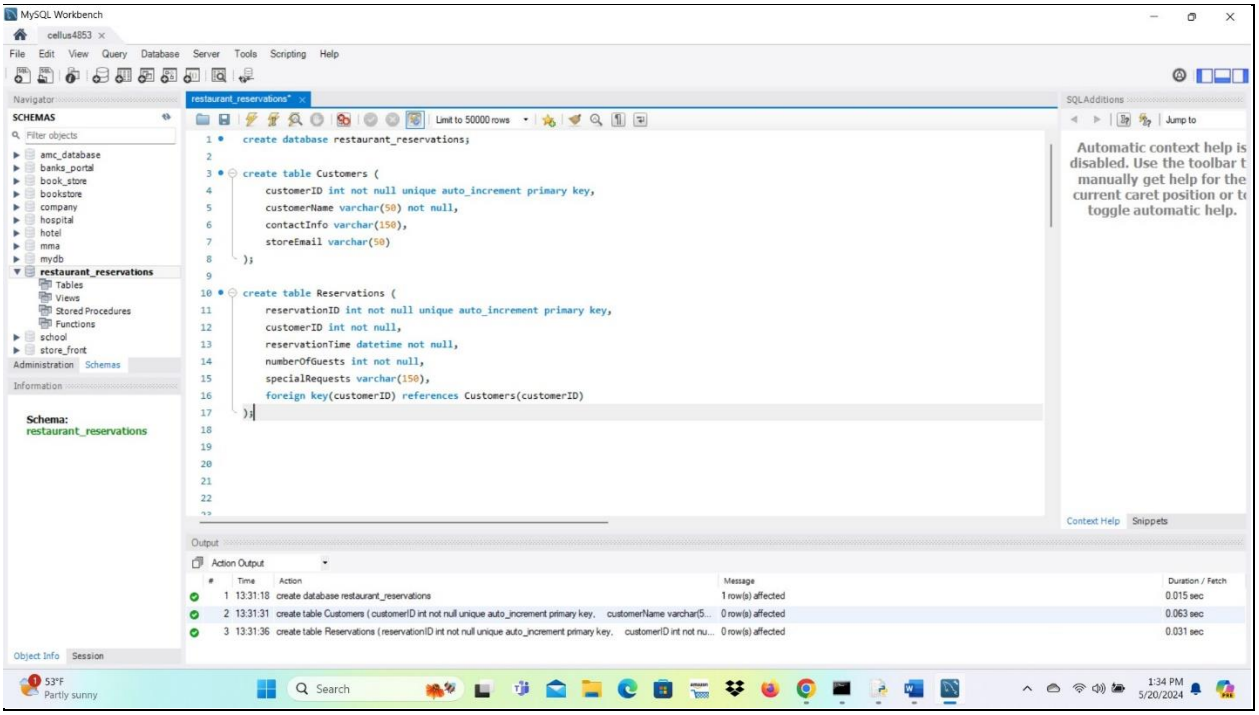
When I got it right it was rewarding. Reviewing the steps helped a lot during this process. Filling in the tables with three tuples took creativity but was fun. I just used family names. Instruction number seven gave me great difficulty. However, I was smart enough to go back to the lessons to complete part of it. The rest came from the website, “MySQL

Connector/Python Developer Guide". Everywhere that I was prompted to input, I would write the word "try", then colon, at the bottom of the mini paragraph "print", "failed to…", and "return". I was very excited after completing this part of the project. After running the code and saw "Port 8000" in blue text, I was eager to see the result. When I did, it was cool to look at and all of the sudden the restaurant menu felt very real. After completing the MySQL portion of the project, I saved it in the file and got it on Github.

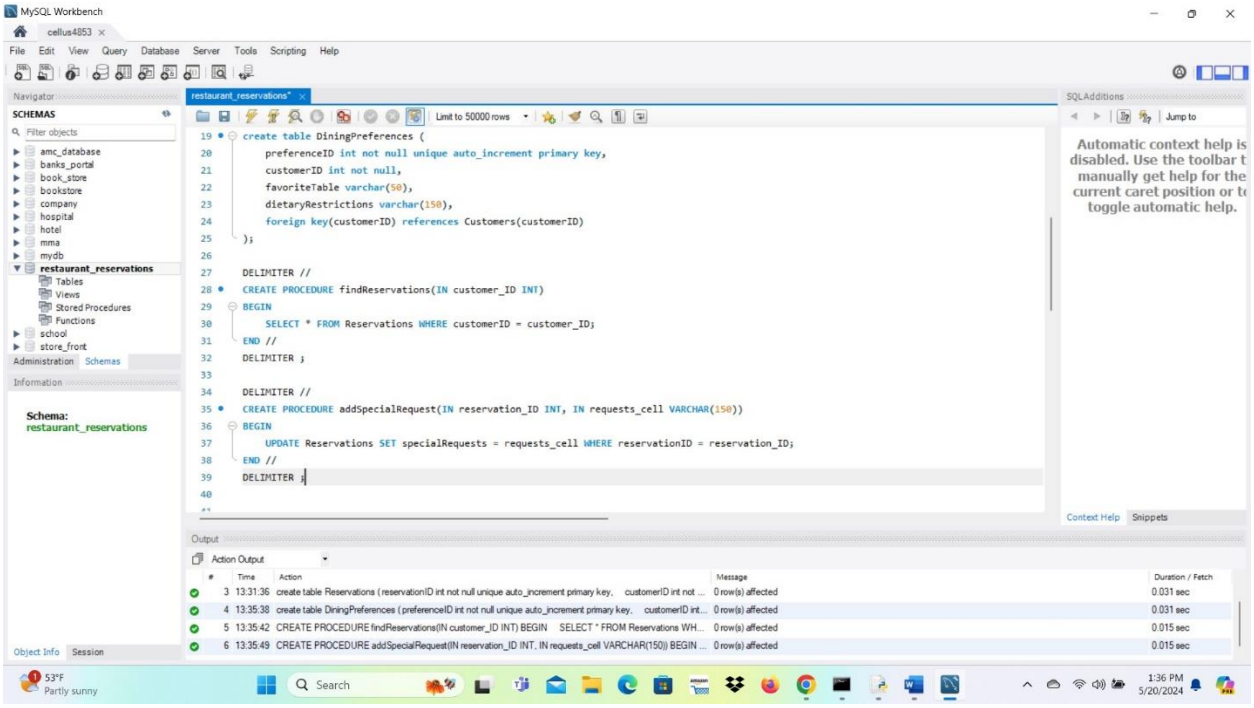
Fortunately for myself, I already have a Github account and it was rather easy to transfer or upload the information onto that platform. I opened my Github account, made certain that the information I transferred over was public, and double checked so as not to make any mistakes. Once I got everything in the repository, I felt that I was near completion. However, it is not over until after the presentation.

Appendix Page

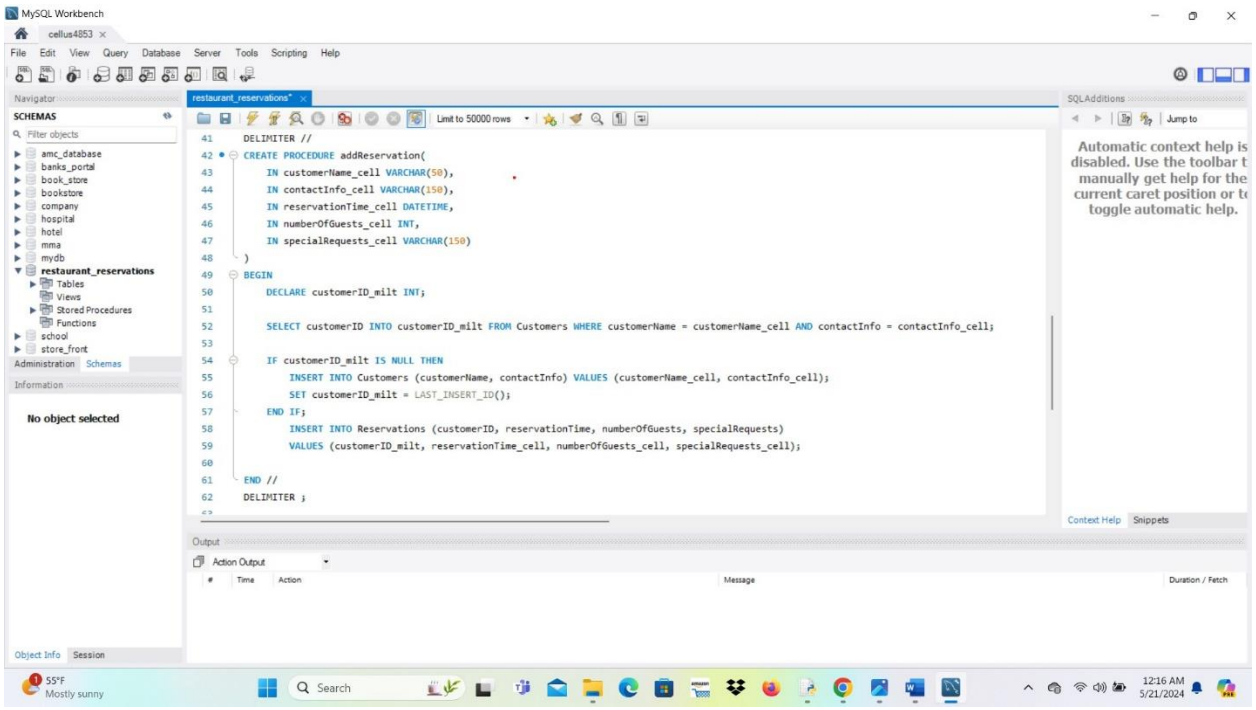
- **Created the database for restaurant_reservations and the first two tables**

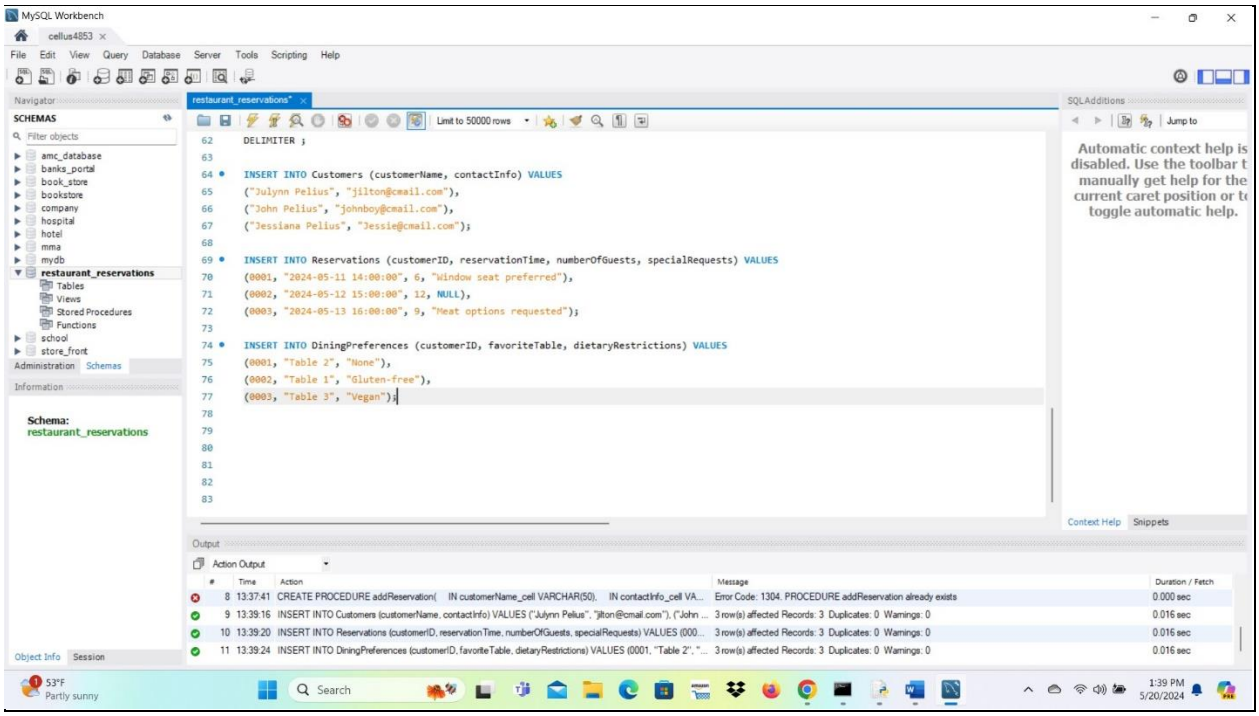


- **Created the third table and completed the first two stored procedures**

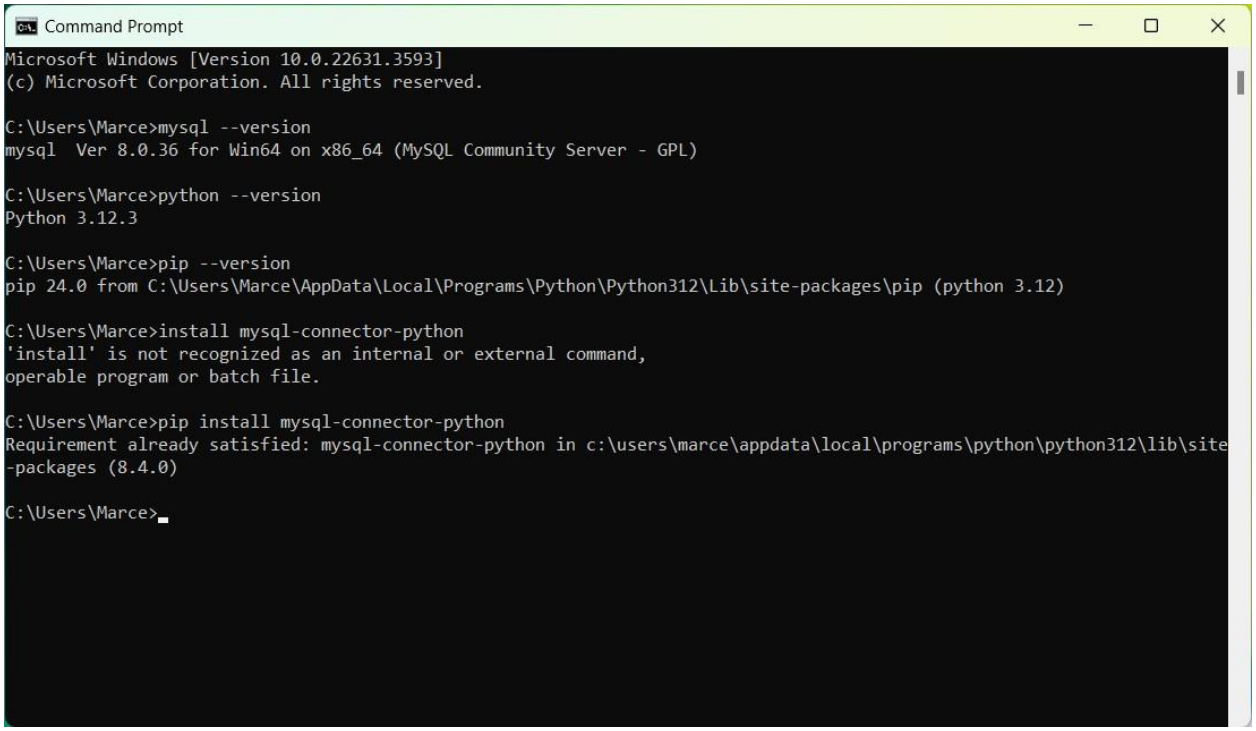


- **Completed the third stored procedure and inserted values into each table**

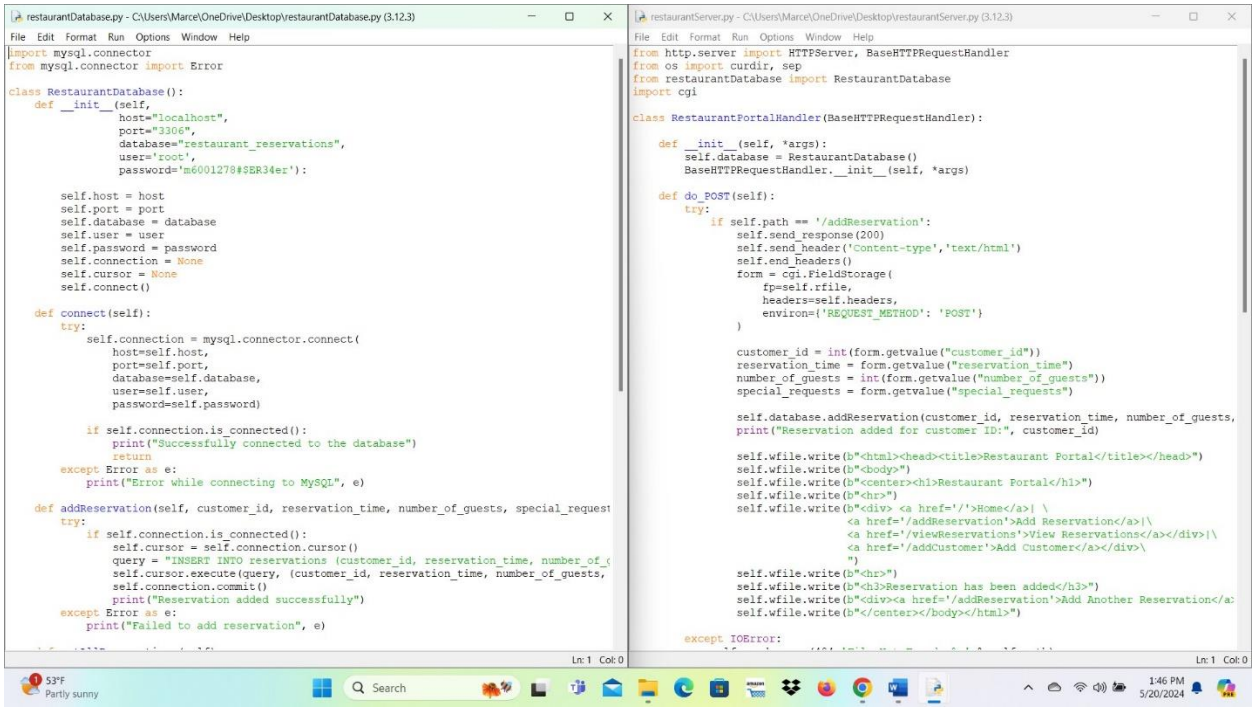




- **Opened Command Prompt to check MySQL, Python, and PIP connection**



- **Opened IDLE (Python) to run modules for restaurantDatabase and restaurantServer**



restaurantDatabase.py - C:\Users\Marce\OneDrive\Desktop\restaurantDatabase.py (3.12.3)

File Edit Format Run Options Window Help

```
import mysql.connector
from mysql.connector import Error

class RestaurantDatabase():
    def __init__(self,
                 host="localhost",
                 port="3306",
                 database="restaurant_reservations",
                 user='root',
                 password='m6001278#SER34er'):

        self.host = host
        self.port = port
        self.database = database
        self.user = user
        self.password = password
        self.connection = None
        self.cursor = None
        self.connect()

    def connect(self):
        try:
            self.connection = mysql.connector.connect(
                host=self.host,
                port=self.port,
                database=self.database,
                user=self.user,
                password=self.password)

            if self.connection.is_connected():
                print("Successfully connected to the database")
                return
        except Error as e:
            print("Error while connecting to MySQL", e)

    def addReservation(self, customer_id, reservation_time, number_of_guests, special_request
        try:
            if self.connection.is_connected():
                self.cursor = self.connection.cursor()
                query = "INSERT INTO reservations (customer_id, reservation_time, number_of_
                self.cursor.execute(query, (customer_id, reservation_time, number_of_guests,
                self.connection.commit()
                print("Reservation added successfully")
        except Error as e:
            print("Failed to add reservation", e)
```

Ln: 1 Col: 0

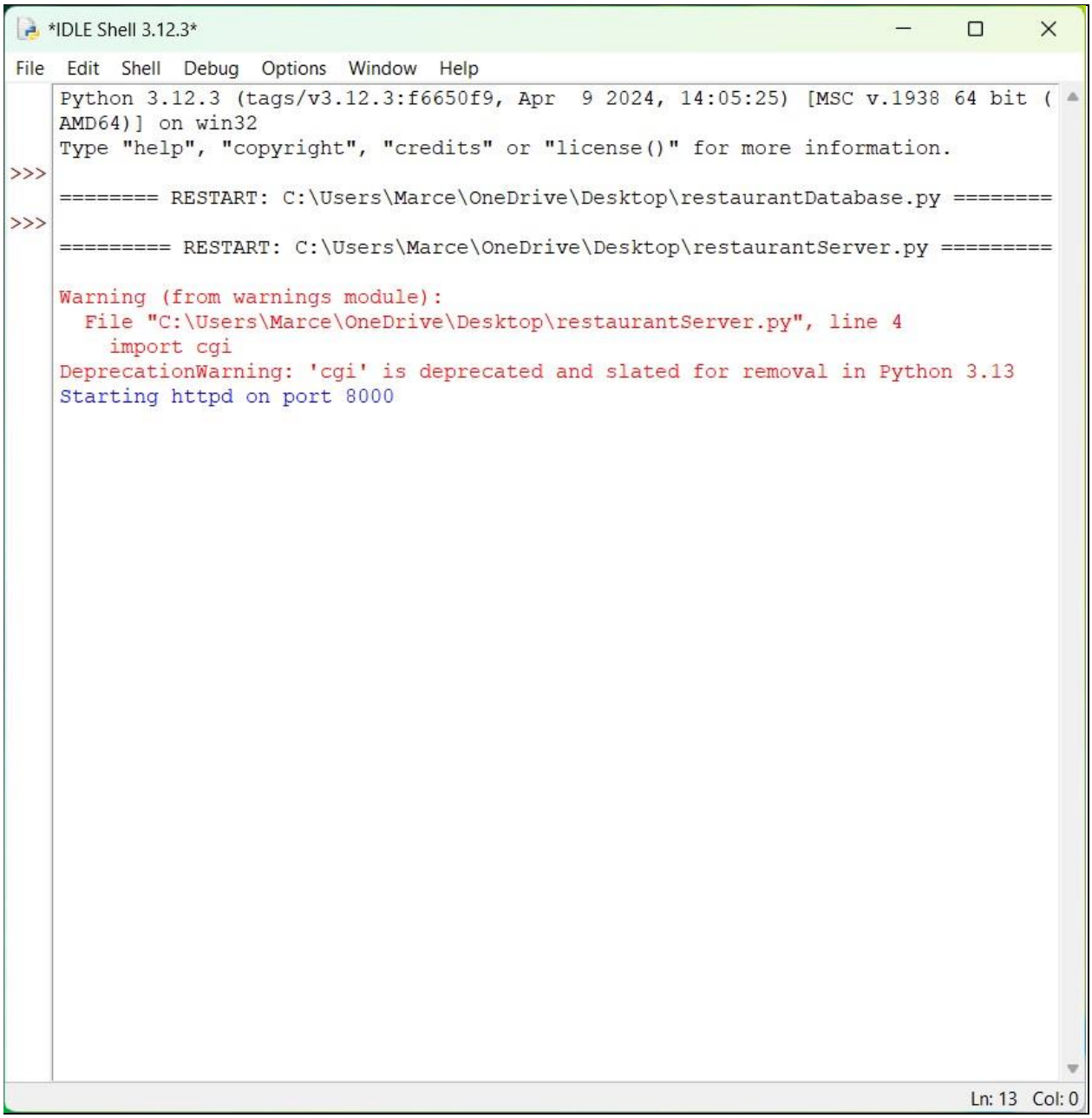
restaurantServer.py - C:\Users\Marce\OneDrive\Desktop\restaurantServer.py (3.12.3)

File Edit Format Run Options Window Help

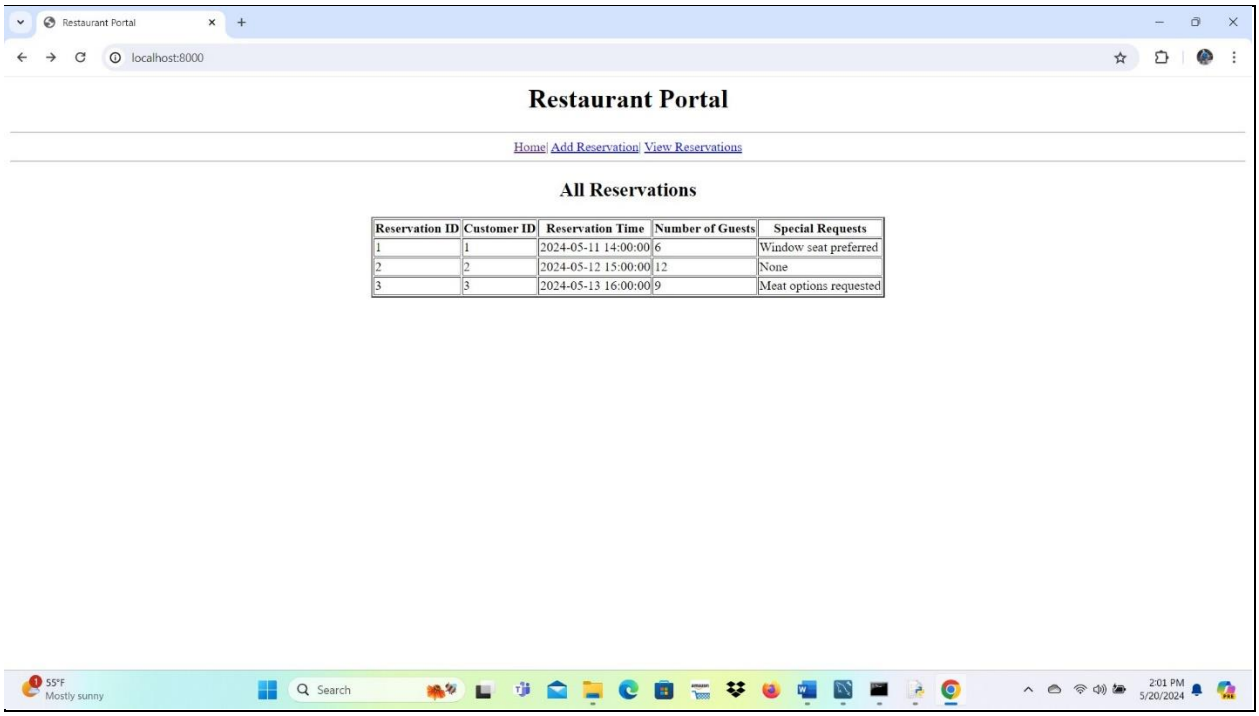
```
from http.server import HTTPServer, BaseHTTPRequestHandler
from os import curdir, sep
from restaurantDatabase import RestaurantDatabase
import cgi

class RestaurantPortalHandler(BaseHTTPRequestHandler):

    def __init__(self, *args):
        self.database = RestaurantDatabase()
        BaseHTTPRequestHandler.__init__(self, *args)

    def do_POST(self):
        try:
            if self.path == '/addReservation':
                self.send_response(200)
                self.send_header('Content-type','text/html')
                self.end_headers()
                form = cgi.FieldStorage(
                    fp=self.rfile,
                    headers=self.headers,
                    environ={'REQUEST_METHOD': 'POST'}
                )

                customer_id = int(form.getvalue("customer_id"))
                reservation_time = form.getvalue("reservation_time")
                number_of_guests = int(form.getvalue("number_of_guests"))
                special_requests = form.getvalue("special_requests")

                self.database.addReservation(customer_id, reservation_time, number_of_guests,
                print("Reservation added for customer ID:", customer_id)

                self.wfile.write(b"<html><head><title>Restaurant Portal</title></head>")
                self.wfile.write(b"<body>")
                self.wfile.write(b"<center><h1>Restaurant Portal</h1>")
                self.wfile.write(b"<hr>")
                self.wfile.write(b"<div> <a href='/'>Home</a>| \
                                   <a href='/addReservation'>Add Reservation</a>|\
                                   <a href='/viewReservations'>View Reservations</a></div>|\
                                   <a href='/addCustomer'>Add Customer</a></div>\
                                   ")
                self.wfile.write(b"<hr>")
                self.wfile.write(b"<h3>Reservation has been added</h3>")
                self.wfile.write(b"<div><a href='/addReservation'>Add Another Reservation</a
                self.wfile.write(b"</center></body></html>")

        except IOError:
```

Ln: 1 Col: 0

- **Ran both modules for the result: Starting httpd on port 8000**



- **Finally, checking my result to satisfaction… "It worked"**


Restaurant Portal
localhost:8000
Restaurant Portal
Home| Add Reservation| View Reservations
All Reservations
Reservation ID | Customer ID | Reservation Time | Number of Guests | Special Requests
1 | 1 | 2024-05-11 14:00:00 | 6 | Window seat preferred
2 | 2 | 2024-05-12 15:00:00 | 12 | None
3 | 3 | 2024-05-13 16:00:00 | 9 | Meat options requested
55°F
Mostly sunny
Search
2:01 PM
5/20/2024