

# Code Review Worksheet

CPSC 491/491L/492L

## Review Identification Information

Date of Review: November 19, 2023

Project Review is Part of: Medcurity

Branch/PR Identifier: PR #12

Code Reviewer(s): Colleen Lemak

Code Author or Source of Code: Brandon Huyck

These following categories should be addressed, though not all might be applicable, it will depend upon the project and code at hand.

### #1 Project, Milestone, Issues, Features Included, and/or Pull Request description

This sprint addresses the initial version of the database API. The script takes values and creates queries to commit them to the DB. Backlog tasks regarding validating data and datatypes have been satisfied. PR #12: "Db connection v1".

### #2 Code Structural, Design Considerations, and Style Guide Suggestions

I like the addition of the config.py file to store credentials in a dictionary which will be defined outside of the script itself. Clean and understandable layout and comments throughout dbapi.py file and will set us up to connect to the DB properly.

### #3 Code Architectural and Efficiency Considerations

Construction of queries seems to be as efficient as possible—I think we should standardize try-catch blocks as you did in data validation because program crashes may be minimized with preemptive data-checking measures. Nice job!

### #4 Execution, Runtime, and Bugs

No issues testing—I assume this code also would fail to run altogether if the connection to the database was faulty (which it is not considering Medcurity's set up).

### #5 Documentation Quality and Completeness

Great progress this week running with the connection to the DB and integration with UI and crawler. This code will act as the middleman to the network tool, so figuring out how data flows and testing each method (in addition to backlog and remaining validation tasks).

## #6 Testing, Tests, Coverage, and suggested testing improvements

No testing implemented yet, next steps include unit tests and more documentation.

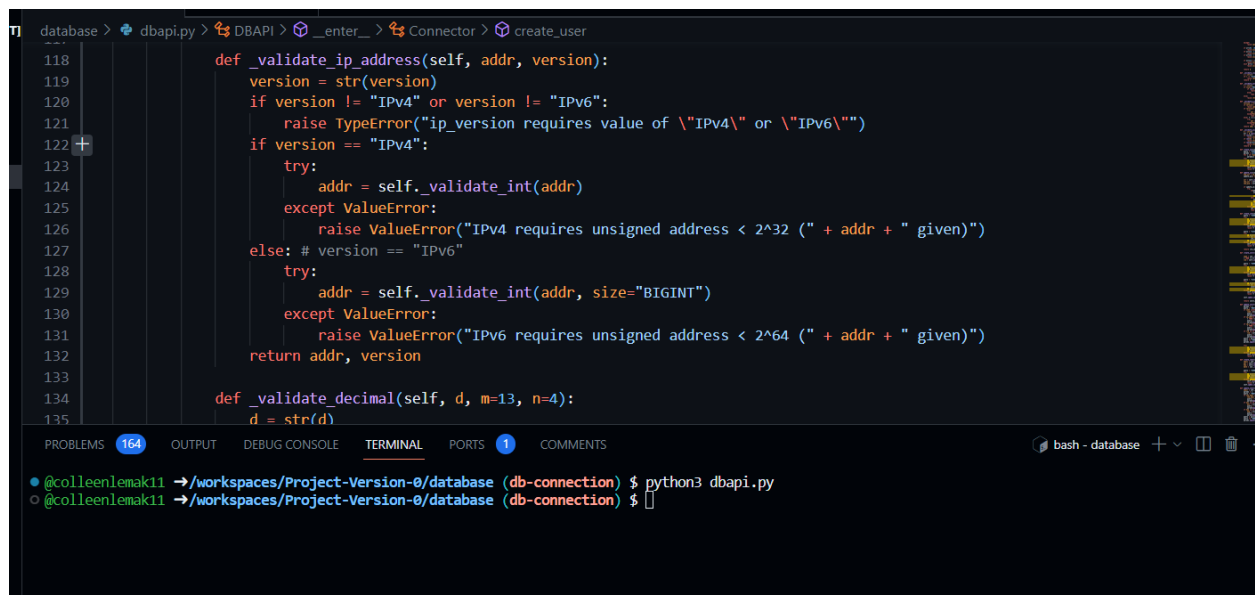
## #7 General Comments, Notes or Other Suggestions for Author

Overall usable and complete work to keep the database progress consistent and measurable.

## #8 Appendix - Include up to five pages of the code under review as a diff.

See git diff or do a diff in VS code/IDE of choice

Show what's being reviewed, code or otherwise (config files, scripts, docs, etc).



The image shows a VS Code editor window with a Python file named `dbapi.py` open. The code defines two methods: `_validate_ip_address` and `_validate_decimal`. The `_validate_ip_address` method checks if the version is "IPv4" or "IPv6" and validates the address accordingly. The `_validate_decimal` method validates a decimal value. The terminal window at the bottom shows the command `python3 dbapi.py` being executed in a shell named `bash - database`.

```
database > dbapi.py > DBAPI > _enter_ > Connector > create_user
118 def _validate_ip_address(self, addr, version):
119     version = str(version)
120     if version != "IPv4" or version != "IPv6":
121         raise TypeError("ip_version requires value of \"IPv4\" or \"IPv6\"")
122     if version == "IPv4":
123         try:
124             addr = self._validate_int(addr)
125         except ValueError:
126             raise ValueError("IPv4 requires unsigned address < 2^32 (\" + addr + \" given)")
127     else: # version == "IPv6"
128         try:
129             addr = self._validate_int(addr, size="BIGINT")
130         except ValueError:
131             raise ValueError("IPv6 requires unsigned address < 2^64 (\" + addr + \" given)")
132     return addr, version
133
134 def _validate_decimal(self, d, m=13, n=4):
135     d = str(d)
```

PROBLEMS 164 OUTPUT DEBUG CONSOLE TERMINAL PORTS 1 COMMENTS

• @colleenlemak11 →/workspaces/Project-Version-0/database (db-connection) \$ python3 dbapi.py  
○ @colleenlemak11 →/workspaces/Project-Version-0/database (db-connection) \$