Metrics:

Total lines of code: 33
Total lines skipped (#nosec): 0

blacklist: Consider possible security implications associated with the subprocess module.
Test ID: B404
Severity: LOW
Confidence: HIGH
CWE: CWE-78
File: .\assignment1_code_sample.py
Line number: 3
More info: https://bandit.readthedocs.io/en/1.8.3/blacklists/blacklist_imports.html#b404-import-subprocess

```
2       import pymysql # type: ignore
3       import subprocess
4       from urllib.request import urlopen
```

start_process_with_partial_path: Starting a process with a partial executable path
Test ID: B607
Severity: LOW
Confidence: HIGH
CWE: CWE-78
File: .\assignment1_code_sample.py
Line number: 22
More info: https://bandit.readthedocs.io/en/1.8.3/plugins/b607_start_process_with_partial_path.html

```
21      def send_email(to, subject, body):
22          subprocess.run(["mail", "-s", subject, to], input=body.encode(), check=True)
23
```

subprocess_without_shell_equals_true: subprocess call - check for execution of untrusted input.
Test ID: B603
Severity: LOW
Confidence: HIGH
CWE: CWE-78
File: .\assignment1_code_sample.py
Line number: 22
More info: https://bandit.readthedocs.io/en/1.8.3/plugins/b603_subprocess_without_shell_equals_true.html

```
21      def send_email(to, subject, body):
22          subprocess.run(["mail", "-s", subject, to], input=body.encode(), check=True)
23
```

blacklist: Audit url open for permitted schemes. Allowing use of file:/ or custom schemes is often unexpected.
Test ID: B310
Severity: MEDIUM
Confidence: HIGH
CWE: CWE-22
File: .\assignment1_code_sample.py
Line number: 27
More info: https://bandit.readthedocs.io/en/1.8.3/blacklists/blacklist_calls.html#b310-urllib-urlopen

```
26          url = 'http://secure-api.com/get-data'
27          data = urlopen(url).read().decode()
28          return data
```

hardcoded_sql_expressions: Possible SQL injection vector through string-based query construction.
Test ID: B608
Severity: MEDIUM
Confidence: LOW
CWE: CWE-89
File: .\assignment1_code_sample.py
Line number: 32
More info: https://bandit.readthedocs.io/en/1.8.3/plugins/b608_hardcoded_sql_expressions.html

```
31      def save_to_db(data):
32          query = f"INSERT INTO mytable (column1, column2) VALUES (%s, %s)"
33          connection = pymysql.connect(**db_config)
```