# Assignment 7:

**By: Kaleb Alstott**

You are a security engineer for a company. Your company is evaluating Bandit (<https://pypi.org/project/bandit/>) to detect any vulnerabilities that have been introduced during development of in-house applications. Because some apps may need exceptions implemented for certain vulnerabilities, while still assessing the rest of the code for other vulnerabilities, you will need to demonstrate not just Bandit’s ability to find vulnerabilites, but also its ability to provide exceptions for vulnerable lines of code. You will demonstrate Bandit’s capabilities by assessing your company’s HR application from earlier.  
  
Note: text in between backticks (`) are commands or code. The backticks should not be included when running the commands or copying code.

## Demonstrating Bandit:

Visit <https://github.com/awzuelsdorf/example_hr_app> , click “Code”, and click “Download Zip”.

Extract the zip file to a new folder.

In the folder, create a new virtual environment.

Activate your virtual environment.

Install the libraries and versions listed in the requirements.txt file in the folder.

Run the following command to install bandit: `pip install bandit==1.7.8` .

Once bandit has been installed, run the command `bandit app.py` and save a screenshot (or multiple screenshots, if needed) of the full output.  
  
Hint: this output should include a string similar to the following, where BXXX would be ‘B’ followed by three positive integers: “Test results:

>> Issue: [BXXX:hardcoded\_sql\_expressions] Possible SQL injection vector through string-based query construction.”

What is the issue id (the ‘BXXX’ value, substituting values for the Xs) in your screenshot(s)?

A screenshot of a computer

Description automatically generated

**Issue id is B608**

## Demonstrating Targeted Exceptions:

Add an exception for the specific issue id you found earlier to the line in app.py that has the SQL injection. For example, if the issue id was ‘B123’, then you would add the comment `# nosec B123` at the end of the line that had the SQL injection. Hint: your screenshots should have a string like “Location:” to determine on what line the SQL injection was found. Do not attempt to fix the SQL injection yourself, just add the exception.

Re-run the `bandit app.py` command from earlier. Save a screenshot (or multiple) of the full output of the command. You should see the following line in your screenshots: “Total potential issues skipped due to specifically being disabled (e.g., #nosec BXXX): 1”

A screenshot of a computer

Description automatically generated

Bandit with --format txt

A computer screen with white text

Description automatically generated

What to submit: A zip file of your screenshots and your modified app.py .