**Tool application report for CS 5103 Course Project:**

**Software Engineering Practice: Word statistics project**

**Mohit Sharma(smw572)**

This project development has used python3 programming language in VS Code editor, and that is the reason, I have used “**pylint**” tool for both code clone detection and static bug detection.

**How to INSTALL and ENABLE pylint in VS Code:**

**pylint** is available as an extension in VS Code editor. Install it from there. Once pylint is enabled we need to follow below steps for the its installation:

* Use the command pip install pylint

A screenshot of a computer program

Description automatically generated with medium confidence

Image: **Screenshot of downloading and installing pylint:**

[**Enable linting**](https://code.visualstudio.com/docs/python/linting#_enable-linting)

* To enable linters, open the Command Palette (Ctrl+Shift+P) and select the Python**: Select Linter** command. The **Select Linter** command adds "python.linting.<linter>Enabled": true to your [settings](https://code.visualstudio.com/docs/getstarted/settings), where <linter> is the name of the chosen linter. See [Specific linters](https://code.visualstudio.com/docs/python/linting#_specific-linters) for details.
* Then use the following settings in "[settings.json](https://code.visualstudio.com/docs/getstarted/settings)":

"python.linting.enabled": true,

"python.linting.pylintEnabled": true,

Enabling a linter prompts you to install the required packages in your selected environment for the chosen linter.A screen shot of a computer

Description automatically generated with low confidence

A screen shot of a computer

Description automatically generated with medium confidence

* Wait for the **pylint** to analyze the code. Any clone detection warnings or errors will appear in the VS Code "Problems" panel.

[**Run linting**](https://code.visualstudio.com/docs/python/linting#_run-linting)

To perform linting, open the Command Palette (Ctrl+Shift+P), filter on "linting", and select **Python: Run Linting**. Linting will run automatically when you save a file.

**Findings of running linting in the project:**

No Cloned code found: Since the code isn’t having any clone code so no specific results for that part, however the code has a few indentation problems , warnings etc. which is reported by the pylint in problems panel.

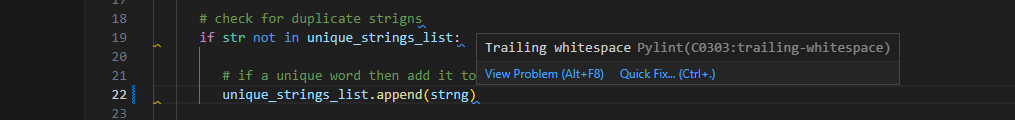
Also, these problems will be visible when you place the cursor on any part of code with colors in errors. Below are a few examples of the said problems reported in the results of this automation tool.

**Unused import report :**

A picture containing text, screenshot, software, multimedia software

Description automatically generated

**Trailing white space:**

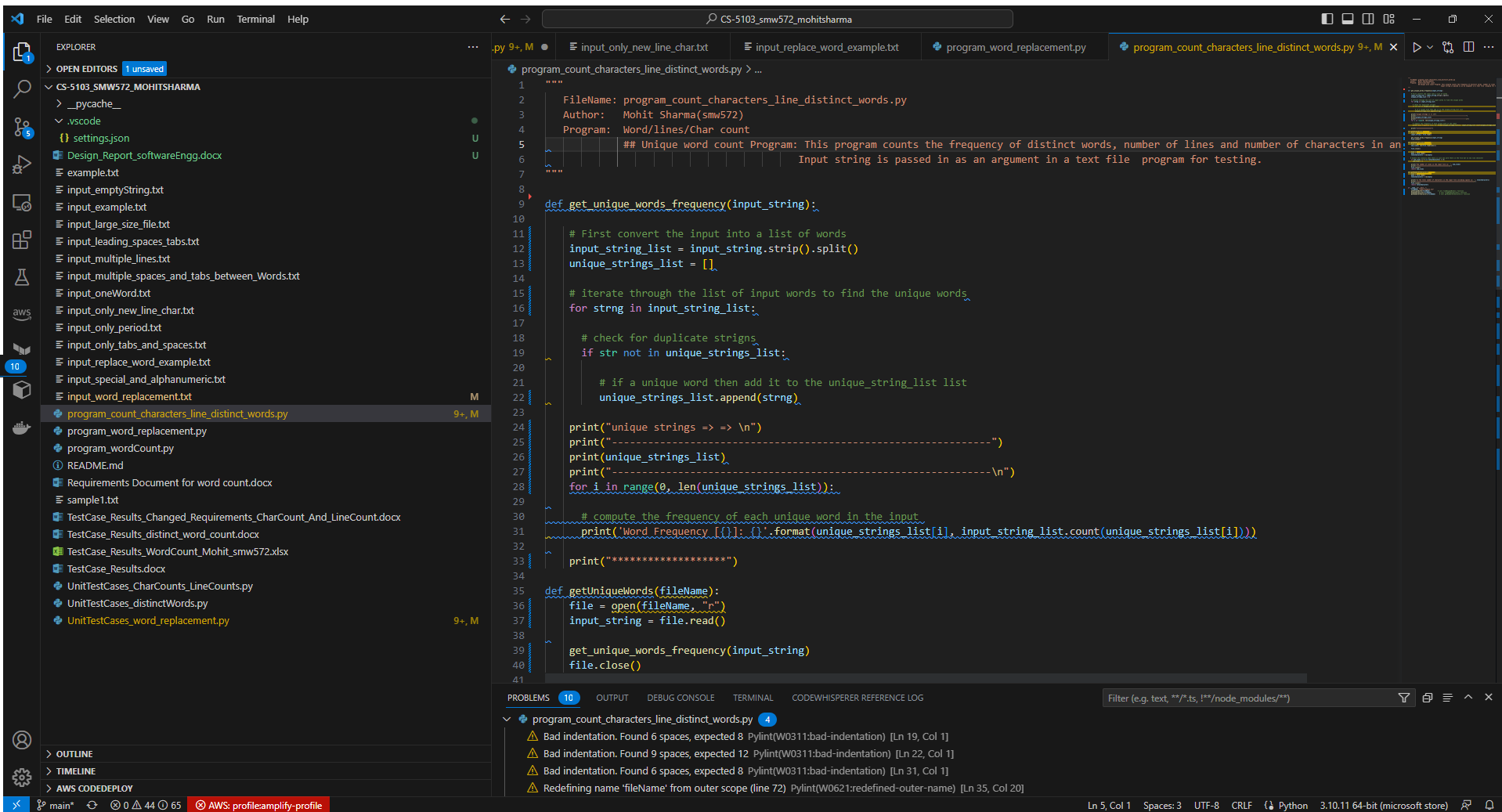
****

**Bad Indentation:**

A screen shot of a computer

Description automatically generated with low confidence

**Results of the code base:**

Graphical user interface, text

Description automatically generated

**Inline comments problem**

Also, a screenshot of the inline comments/problems popping up when the cursor is placed over the problem part of the code

**Experience** The experience of using the pylint, an automated tool, has been very learning one.

* Automated tools are quite flexible , easy to use and learn.
* Self-explanatory sometimes for an experienced programmer.
* Provide a fast pace of development as a developer could see the errors and warnings etc.
* Let the developer know what are the linting problems, right at the same time when coding before compiling or running the code, which is a good time saving component.
* These kinds of tools come in handy when the project is quite big with thousands of lines of code and in extreme programing environments. Where a developer could identify the problems quickly without testing sometimes.

**Overall, it is a great learning and time saving to use the automated tools.**