Name: Chidi Uchechukwu
Course: Python Main Project
Date: April 08, 2024

This project ensures your spreadsheet is clear of inadequacies and inconsistencies. With this simple Python project, you can automate the process of identifying and correcting data irregularities, specifically focusing on email address formats. You would even generate reports on what needs manual review.

First of all, make sure to replace "your_spreadsheet.xlsx" with the path to the Excel file you intend to clean up. It's essential that this spreadsheet has email addresses located in the second column for the script to correctly identify and process them.

Once you run the script, it diligently begins to read each row, starting from the second one (assuming the first row serves as headers). It meticulously checks the format of each email address. If it encounters any that don't conform to standard email formats, it doesn't just flag them; it tries to fix common errors on the spot. For instance, if an email address mistakenly contains spaces, the script automatically replaces them with dots and corrects the capitalization issues.

By the end of its operation, the script produces two key outputs:

A Corrected Spreadsheet: Named "corrected_data.xlsx", this file contains all the data from your original spreadsheet, with the added benefit of having had incorrect email addresses automatically corrected.
An Issues Report: This Word document, titled "issues_report.docx", details every instance where the script encountered email addresses that it couldn't auto-correct, pinpointing exactly where you need to look to make manual corrections.

```
Name:   Chidi Uchechukwu
Course: Python Main Project
Date:   April 08, 2024


import re
from openpyxl import load_workbook
from docx import Document

# Email validation pattern
email_pattern = re.compile(r"^[a-z0-9]+[\._]?[a-z0-9]+[@]\w+[.]\w{2,3}$")

def validate_email(email):
    """Check if the email is in a valid format."""
    return re.match(email_pattern, email)

def correct_email(email):
    """Attempt to correct common email format mistakes."""
    # Example correction: replace spaces with dots
    corrected_email = email.replace(" ", ".").lower()
    return corrected_email if validate_email(corrected_email) else email

def process_spreadsheet(file_path):
    wb = load_workbook(filename=file_path)
    ws = wb.active

    issues_report = Document()
    issues_report.add_heading('Data Issues Report', 0)

    for row in ws.iter_rows(min_row=2, values_only=True):
        email = row[1]  # Assuming email is in the second column
        valid_email = validate_email(email)

        if not valid_email:
            corrected_email = correct_email(email)
            if email != corrected_email:
                # Auto-corrected email
                print(f"Auto-corrected {email} to {corrected_email}")
                ws.cell(row=row[0].row, column=2, value=corrected_email)
            else:
                # Report issue if not corrected
                issues_report.add_paragraph(f"Row {row[0].row}: Invalid email
{email}")

    wb.save("corrected_data.xlsx")
    issues_report.save("issues_report.docx")
```