**Demonstration2**

**import ast**

**import re**

**def check\_functions(source\_code):**

**issues = []**

**tree = ast.parse(source\_code)**

**for node in ast.walk(tree):**

**if isinstance(node, ast.FunctionDef):**

**func\_name = node.name**

**docstring = ast.get\_docstring(node)**

**if not docstring:**

**issues.append(f"Function '{func\_name}' is missing a docstring.")**

**if not re.match(r'^[a-z\_][a-z0-9\_]\*$', func\_name):**

**issues.append(f"Function name '{func\_name}' does not follow snake\_case convention.")**

**return issues**

**def check\_indentation(lines):**

**issues = []**

**for i, line in enumerate(lines, start=1):**

**if line.strip() == "":**

**continue**

**leading\_spaces = len(line) - len(line.lstrip(' '))**

**if leading\_spaces % 4 != 0 and line.startswith(' '):**

**issues.append(f"Line {i}: Indentation is not a multiple of 4 spaces.")**

**return issues**

**def run\_sqa\_checklist(file\_path):**

**try:**

**with open(file\_path, 'r') as f:**

**code = f.read()**

**lines = code.split('\n')**

**print(f"Checking '{file\_path}' for SQA compliance...\n")**

**func\_issues = check\_functions(code)**

**indent\_issues = check\_indentation(lines)**

**all\_issues = func\_issues + indent\_issues**

**if not all\_issues:**

**print(" Code passed all SQA checks!")**

**else:**

**print(" Issues found:")**

**for issue in all\_issues:**

**print("-", issue)**

**except FileNotFoundError:**

**print("Error: File not found.")**

**except Exception as e:**

**print(f"Error during analysis: {e}")**

**if \_\_name\_\_ == "\_\_main\_\_":**

**filename = input("Enter the Python file path to check (e.g., filename.py): ").strip()**

**run\_sqa\_checklist(filename)**

**sample.py**

**def greet\_user(name):**

**"""Prints a greeting message to the user."""**

**print(f"Hello, {name}!")**

**def calculate\_sum(a, b):**

**"""Returns the sum of two numbers."""**

**return a + b**

**def multiply(x, y):**

**"""Returns the product of two numbers."""**

**return x \* y**

**def is\_even(number):**

**"""Checks if a number is even."""**

**return number % 2 == 0**