|  |  |  |
| --- | --- | --- |
| **Author** | **Reviewer** | **Approver** |
| BaoHT5 |  |  |

**Exercise: Enhancing Code Quality with CodeVista**

## **1. Objectives**

* Understand how to leverage CodeVista for refactoring code and generating documentation.
* Apply CodeVista's features to improve code readability and maintainability.
* Experience the integration of AI tools in the development workflow.

## **2. Problem Statement**

Maintaining clear documentation and clean code is essential for efficient software development.

This assignment challenges you to use CodeVista to generate documentation for a given function and refactor a complex function to enhance its readability and structure, demonstrating the practical value of AI-assisted development in real-world scenarios.

## **3. Inputs / Shared Artifacts**

* **CodeVista Access:** Ensure you have CodeVista installed and configured in your development environment.
* **Sample Code:**  
   Use ChatGPT to generate a function - Spelling Corrector
* **Complex Function:**  
   Use ChatGPT to generate a more complex function Binary Image Classifier (Hardcoded Logic).
* *No additional starter code, datasets, or templates are provided. You must create your own mock data and code as needed.*

## **4. Expected Outcomes**

* **Documentation Generation:**
  + Use CodeVista to generate documentation for the sample function.
  + Review and, if necessary, edit the generated documentation to ensure clarity and completeness.
* **Code Refactoring:**
  + Use CodeVista to refactor the complex function, breaking it down into smaller, manageable functions or improving its structure.
  + Ensure the refactored code maintains the original functionality.
* **Reflection:**
  + Write a brief summary of your experience using CodeVista for these tasks, noting any challenges and how the tool assisted in overcoming them.

## **5. Concepts Covered**

* **AI-Assisted Documentation:** Generating clear and concise documentation using AI tools.
* **Code Refactoring:** Improving code structure and readability without altering its functionality.
* **Integration of AI in Development:** Understanding how AI tools like CodeVista can enhance the development process.

## **6. Example: Step-by-Step Instructions**

**Step 1: Generate Sample Code with ChatGPT**

* Prompt ChatGPT to create a function that calculates the average grade of a list of students.

def calculate\_average(grades):

total = sum(grades)

return total / len(grades)

**Step 2: Generate a Complex Function with ChatGPT**

* Ask ChatGPT to generate a function that processes a dictionary of student names and their grade lists, calculates the average for each, and determines the top student.

def find\_top\_student(student\_grades):

top\_student = None

top\_average = 0

for student, grades in student\_grades.items():

avg = sum(grades) / len(grades)

if avg > top\_average:

top\_average = avg

top\_student = student

return top\_student, top\_average

**Step 3: Documentation Generation with CodeVista**

* Use CodeVista’s documentation feature to auto-generate docstrings for calculate\_average.
* Review and edit the generated documentation to ensure it explains parameters, return values, and edge cases.

**Step 4: Refactor the Complex Function with CodeVista**

* Use CodeVista’s refactoring feature to break find\_top\_student into:
  + calculate\_average(grades)
  + find\_top\_student(student\_grades)
* Ensure the new structure is more readable and all original functionality is preserved.

**Step 5: Reflection**

* Write a short summary (5–7 sentences) on your experience:
  + How did CodeVista’s AI-powered features help?
  + What improvements did you notice in documentation and code clarity?
  + Any limitations or challenges?

## **7. Final Submission Checklist**

* **Screen Recording (max 5 minutes):**
  + Demonstrate the process:
    - Generating and editing documentation for a function
    - Refactoring a complex function for readability
* **Supporting Files:**
  + Source code with before/after examples
  + Documentation generated/edited by CodeVista
  + Short summary (README or PDF) including:
    - Example ChatGPT prompts used
    - Description of the process, challenges, and how CodeVista helped (Optional)
    - Key takeaways about AI in your development workflow