|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AssignmentNumber:9.1**(Presentassignmentnumber)/**24**(Totalnumberofassignments) NAME:- H.V.S.VYSHNAVI  ROLLNO:-2403A51182 | | | | |
|  | **Q.No.** | **Question** | | ***Expected Time***  ***to***  ***complete*** |
|  | 1 |  | **Lab9–DocumentationGeneration:AutomaticDocumentationand Code Comments**  **LabObjectives**   * TouseAI-assistedcodingtoolsforgeneratingPython documentation and code comments. * Toapplyzero-shot,few-shot,andcontext-basedprompt engineering for documentation creation. * Topracticegeneratingandrefiningdocstrings,inlinecomments, and module-level documentation. * Tocompareoutputsfromdifferentpromptingstylesforquality analysis. | Week5- Monday |
| **TaskDescription#1**(Documentation–Google-StyleDocstringsfor Python Functions)   * Task:UseAItoaddGoogle-styledocstringstoallfunctionsina given Python script. * Instructions:   + PromptAItogeneratedocstringswithoutprovidingany input-output examples.   + Ensureeachdocstringincludes:     - Functiondescription     - Parameterswithtypehints     - Returnvalueswithtypehints     - Exampleusage   + Reviewthegenerateddocstringsforaccuracyand formatting. |

|  |  |  |
| --- | --- | --- |
|  | **Prompt:**  IhaveaPythonscript containingseveralfunctions.Pleaseadd **Google- style docstrings** to **all functions** in the script.  Thedocstringsshouldincludethe followingelements:   * A**briefdescription** ofwhatthe functiondoes. * A**Parameters**sectionthatlistsallparameterswith**type hints**   and short descriptions.   * A**Returns**sectionthat liststhe**returntype**andabrief description. * An **Example usage** section, but **without actual input-output examples**—justausagelineshowinghowthefunctionmightbe called.   Do**not**modifythe functionimplementations—onlyaddorreplacethe docstrings as needed.  **CODE:-** |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **OUTPUT:-**    **OBSERVATIONS:-** |  |
| **TaskDescription#2**(Documentation–InlineCommentsforComplex Logic)   * Task:UseAItoaddmeaningfulinlinecommentstoaPython program explaining only complex logic parts. * Instructions:   + ProvideaPythonscriptwithoutcommentstotheAI.   + InstructAItoskipobvioussyntaxexplanationsandfocus only on tricky or non-intuitive code sections.   + Verifythatcommentsimprovecodereadabilityand maintainability. |

|  |  |  |
| --- | --- | --- |
|  | **CODE:-** |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | **OUTPUT:-**    **OBSERVATIONS:-** |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |

**TaskDescription#3**(Documentation–Module-LevelDocumentation)

* Task:UseAI tocreateamodule-leveldocstringsummarizing the
* purpose,dependencies,andmainfunctions/classesofaPython file.
* Instructions:
  + SupplytheentirePython filetoAI.
  + InstructAIto writeasingle multi-linedocstringatthetop of the file.
  + Ensurethedocstringclearlydescribesfunctionalityand usage without rewriting the entire code.
* ExpectedOutput#3:
  + Acomplete,clear,andconcisemodule-leveldocstringat the beginning of the file.

# PROMPT:-

Iamproviding youwithanentirePythonfile.Pleaseadda\*\*module- level docstring\*\* at the very top of the file.

Thedocstringshouldbeasinglemulti-linestringthatincludes:

* Aconcisesummaryofthemodule’spurpose.
* Keydependenciesor importsifapplicable.
* Mainfunctionsorclassesincludedinthe module.
* Basicusagenotesorhowthismodule might beused(brief,not afull tutorial).

Do\*\*not\*\*rewriteorexplaintheentirecode —just provideaclear and professional summary suitable for the top of a Python file.

ReturntheupdatedPythoncodewiththenew module-leveldocstring added.

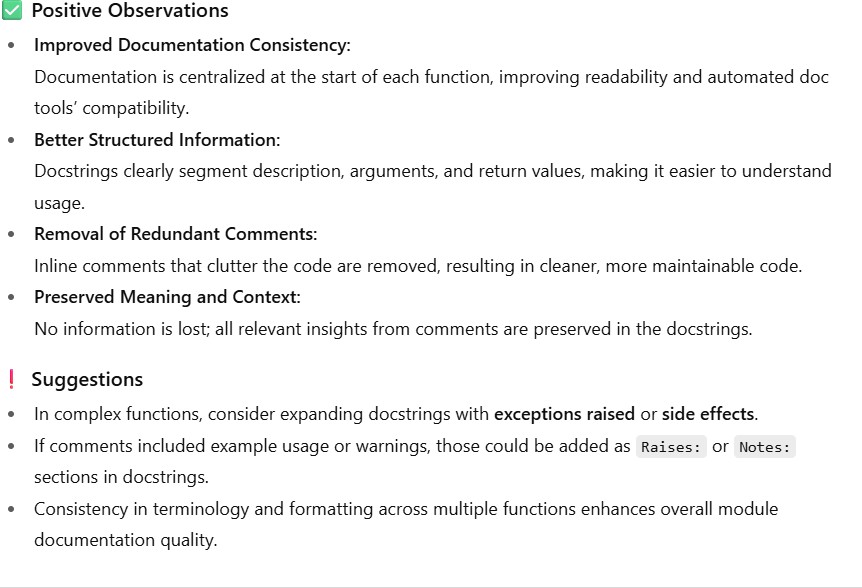
|  |  |  |
| --- | --- | --- |
|  | **CODE:-** |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **OUTPUT:-**    **OBSERVATIONS:-** |  |
| **TaskDescription#4**(Documentation–ConvertCommentsto Structured Docstrings)   * Task:UseAItotransformexistinginlinecommentsinto structured function docstrings following Google style. * Instructions:   + ProvideAIwithPythoncodecontaininginlinecomments.   + AskAItomoverelevantdetailsfromcommentsinto function docstrings.   + Verifythatthenewdocstringskeepthemeaningintact while improving structure. |

|  |  |  |
| --- | --- | --- |
|  | **PROMPT:-**  Iamproviding youwithaPythonscriptthatcontainsinlinecomments inside functions.  Pleasetransformtheseinlinecommentsintowell-structured\*\*Google- style docstrings\*\* for each function, moving all relevant information from the comments into the docstrings.  Makesurethedocstringsinclude:   * Aconcisefunctiondescription. * Parameterdescriptionswithtypehints. * Returntypeanddescription(ifapplicable). * Anyotherimportantinformationpreviouslypresentinthecomments. Remove the inline comments once they are moved to the docstrings.   Returnthe updatedPythoncode withthe new docstrings.  **CODE:-** |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  | **OUTPUT:-**    **OBSERVATIONS:-** |  |



|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |

**TaskDescription#5**(Documentation–ReviewandCorrect Docstrings)

* Task:UseAItoidentifyandcorrectinaccuraciesinexisting docstrings.
* Instructions:
  + ProvidePythoncodewithoutdatedorincorrect docstrings.
  + InstructAItorewriteeachdocstringto matchthecurrent code behavior.
  + EnsurecorrectionsfollowGoogle-style formatting.
* ExpectedOutput#5:
  + Pythonfilewithupdated,accurate,andstandardized docstrings.

# PROMPT:-

I’mprovidingaPythonscript wheresome functionshaveoutdatedor incorrect docstrings.

Pleasecarefullyreviewand\*\*rewriteeachdocstring\*\*sothat it accurately reflects the function’s current behavior.

Followthe\*\*Google-style\*\*docstringformat,andensurethat:

-Thefunctiondescriptioniscorrect.

|  |  |  |
| --- | --- | --- |
|  | * Parametersand returntypesareaccurateandfullydescribed. * Anyremovedorchangedfunctionalityisnolonger referenced. * Theformatting iscleanandconsistent.   Do notchangethefunctioncode —onlycorrectthedocstrings. Return the updated Python code.  **CODE:-** |  |

|  |  |  |
| --- | --- | --- |
|  | **OUTPUT:-** |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | **OBSERVATIONS:-** |  |
| **TaskDescription#6**(Documentation–PromptComparison Experiment)   * Task:Comparedocumentationoutputfromavagueprompt anda detailed prompt for the same Python function. * Instructions:   + Create two prompts: one simple (“Add comments to this function”)andonedetailed(“AddGoogle-styledocstrings with parameters, return types, and examples”).   + UseAIto processthesamePythonfunctionwithboth prompts.   + Analyzeandrecorddifferencesinquality,accuracy,and completeness. |

|  |  |  |
| --- | --- | --- |
|  | **PROMPTS:-**  **🔹VaguePrompt**  Addcommentstothisfunction.  **🔹DetailedPrompt**  PleaseaddastructuredGoogle-styledocstringtothefollowingPython function. Make sure to include:   * Aconcisedescriptionofthefunction’spurpose * Parameterswithtypehintsand descriptions * Returntypeanditsdescription * Anexampleusage(withoutactualoutputvalues) UseproperformattingforGoogle-styledocstrings   **CODE:-**  PYTHONFUNCTIONCODE:    **CODE FORVAGUEPROMPT:** |  |

|  |  |  |
| --- | --- | --- |
|  | **CODEFORDETAILED PROMPT:-**    **OUTPUT:-** |  |

|  |  |  |
| --- | --- | --- |
|  | **OBSERVATIONS:-** |  |