NAME:CHANDA HARINI ID:2403A510E1 BATCH:05

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCHOOLOFCOMPUTERSCIENCEANDARTIFICIAL INTELLIGENCE** | | | | | **DEPARTMENTOFCOMPUTERSCIENCE ENGINEERING** | | | | |
| **ProgramName:**B.Tech | | | | **AssignmentType:Lab** | | | **AcademicYear:**2025-2026 | | |
| **CourseCoordinatorName STUDENT DETAILS:** | | | | VenkataramanaVeeramsetty.  CHANDA HARINI  Batch-05  2403a510E1 | | | | | |
| **Instructor(s)Name** | | | | Dr.V.Venkataramana(Co-Ordinator) | | | |  | |
| Dr.T.SampathKumar | | | |
| Dr.PramodaPatro | | | |
| Dr.BrijKishor Tiwari | | | |
| Dr.J.Ravichander | | | |
| Dr.MohammandAliShaik | | | |
| Dr.AnirodhKumar | | | |
| Mr.S.Naresh Kumar | | | |
| Dr.RAJESHVELPULA | | | |
| Mr.KundhanKumar | | | |
| Ms.Ch.Rajitha | | | |
| Mr.MPrakash | | | |
| Mr.B.Raju | | | |
| Intern1(Dharmateja) | | | |
| Intern2(Sai Prasad) | | | |
| Intern3(Sowmya) | | | |
| NS\_2( Mounika) | | | |
| **CourseCode** | | | 24CS002PC215 | **CourseTitle** | | AIAssistedCoding | | | |
| **Year/Sem** | | | II/I | **Regulation** | | R24 | | | |
| **DateandDay**  **of Assignment** | | | Week1- Monday | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicableto Batches** | | 24CSBTB01To 24CSBTB39 | | | |
| **AssignmentNumber:2.1**(Presentassignmentnumber)/**24**(Totalnumberofassignments) | | | | | | | | | |
|  | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | | ***Expected Time***  ***to***  ***complete*** |
|  | 1 | Lab 1: Environment Setup – GitHub Copilot and VS Code Integration Lab Objectives: ● To install and configure GitHub Copilot in Visual Studio Code.  To explore AI-assisted code generation using GitHub Copilot. ● To analyze the accuracy and effectiveness of Copilot's code suggestions. ● To understand prompt-based programming using comments and code context Lab Outcomes (LOs): After completing this lab, students will be able to: ● Set up GitHub Copilot in VS Code successfully. ● Use inline comments and context to generate code with Copilot. ● Evaluate AI-generated code for correctness and readability. ● Compare code suggestions based on different prompts and programming styles. Task 0 ● Install and configure GitHub Copilot in VS Code. Take screenshots of each step. Expected Output ● Install and configure GitHub Copilot in VS Code. Take screenshots of each step.  **Task 1:** Factorial without Functions ● Description: Use GitHub Copilot to generate a Python program that calculates the factorial of a number without defining any functions (using loops directly in the main code). ● Expected Output: o A working program that correctly calculates the factorial for user-provided input. o Screenshots of the code generation process.  **Prompt:**    **Code:**    **Output:**    **Task 2**: Improving Efficiency ● Description: Examine the Copilot-generated code from Task 1 and demonstrate how its efficiency can be improved (e.g., removing unnecessary  variables, optimizing loops). ● Expected Output: o Original and improved versions of the code. o Explanation of how the improvements enhance performance  **prompt and code:**      **Task 3**: Factorial with Functions ● Description: Use GitHub Copilot to generate a Python program that calculates the factorial of a number using a user-defined function. ● Expected Output: o Correctly working factorial function with sample outputs. o Documentation of the steps Copilot followed to generate the function.  **Prompt and code:**      **Task 4:** Comparative Analysis – With vs Without Functions ● Description: Differentiate between the Copilot-generated factorial program with functions and without functions in terms of logic, reusability, and execution. ● Expected Output: o A comparison table or short report explaining the differences      **Task 5**: Iterative vs Recursive Factorial ● Description: Prompt GitHub Copilot to generate both iterative and recursive versions of the factorial function. ● Expected Output: o Two correct implementations. o A documented comparison of logic, performance, and execution flow between iterative and recursive approaches.  **Prompt:**    **Code:**    **Output:**    **Submission Requirements** 1. Generate code for each task with comments. 2. Screenshots of Copilot suggestions. 3. Comparative analysis reports (Task 4 and Task 5). 4. Sample inputs/outputs demonstrating correctness. Note: Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots  **Evaluation Criteria:**  Criteria Max Marks Successful Setup of Copilot 0.5 Comparative Analysis – With vs Without Functions 1 Iterative vs Recursive Factorial 1 Total 2.5 Marks | | | | | | | Week1- Monday |

|  |  |  |
| --- | --- | --- |
|  | CursorAI  **LabObjectives:**   * ToexploreandevaluatethefunctionalityofGoogleGeminifor AI-assisted coding within Google Colab. * TounderstandanduseCursorAIforcodegeneration, explanation, and refactoring. * TocompareoutputsandusabilitybetweenGemini,GitHub Copilot, and Cursor AI. * ToperformcodeoptimizationanddocumentationusingAI tools.   **LabOutcomes (LOs):**  Aftercompletingthislab,studentswillbeable to:   * GeneratePythoncodeusingGoogleGemini inGoogleColab. * Analyzetheeffectivenessofcodeexplanationsandsuggestions by Gemini. * SetupanduseCursorAI forAI-poweredcoding assistance. * EvaluateandrefactorcodeusingCursorAIfeatures. * CompareAItoolbehaviorandcodequalityacrossdifferent platforms. |  |
|  | TaskDescription#1   * Use Google Gemini in Colab to write a Python function that readsalistofnumbersandcalculatesthemean,minimum,and maximum values.   ExpectedOutput#1   * Functionalcodewithcorrectoutputandscreenshot. |  |