Name : R.prajwal Hall Ticket No: 2303A51830 Batch: 26

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE** | | | | | **DEPARTMENT OF COMPUTER SCIENCE ENGINEERING** | | | | |
| **Program Name:** B. Tech | | | | **Assignment Type: Lab** | | | **Academic Year:**2025-2026 | | |
| **Course Coordinator Name** | | | | Dr. Rishabh Mittal | | | | | |
| **Instructor(s) Name** | | | | |  | | --- | | Mr. S Naresh Kumar | | Ms. B. Swathi | | Dr. Sasanko Shekhar Gantayat | | Mr. Md Sallauddin | | Dr. Mathivanan | | Mr. Y Srikanth | | Ms. N Shilpa | | Dr. Rishabh Mittal (Coordinator) | | Dr. R. Prashant Kumar | | Mr. Ankushavali MD | | Mr. B Viswanath | | Ms. Sujitha Reddy | | Ms. A. Anitha | | Ms. M.Madhuri | | Ms. Katherashala Swetha | | Ms. Velpula sumalatha | | Mr. Bingi Raju | | | | | | |
| **CourseCode** | | | 23CS002PC304 | **Course Title** | | AI Assisted Coding | | | |
| **Year/Sem** | | | III/II | **Regulation** | | R23 | | | |
| **Date and Day**  **of Assignment** | | | **Week2** | **Time(s)** | | 23CSBTB01 To 23CSBTB52 | | | |
| **Duration** | | | 2 Hours | **Applicable to**  **Batches** | | All batches | | | |
| **Assignment Number: 3.4** (Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***Expected Time***  ***to complete*** |  |
|  | 1 | Lab 4: Advanced Prompt Engineering – Zero-shot, One-shot, and Few-shot Techniques  **Task 1: Zero-shot Prompt – Fibonacci Series Generator**  **Task Description #1**  • Without giving an example, write a single comment prompt asking GitHub Copilot to generate a Python function to print the first N Fibonacci numbers.  **Expected Output #1**  **•** A complete Python function generated by Copilot without any example provided.  • Correct output for sample input N = 7 ➝ 0 1 1 2 3 5 8  • Observation on how Copilot understood the instruction with zero context.  **1. Fibonacci Function (Earlier)**   * **Purpose**: Print the first N Fibonacci numbers * **Function**: [print\_fibonacci(n)](vscode-file://vscode-app/c:/Users/prade/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html" \o ") * **Logic**: Uses two variables (a, b) to track consecutive Fibonacci numbers, iterates N times * **Example**: print\_fibonacci(10) prints: 0 1 1 2 3 5 8 13 21 34   **Task 2: One-shot Prompt – List Reversal Function**  **Task Description #2**  • Write a comment prompt to reverse a list and provide one example below the comment to guide Copilot.  **Expected Output #2**  • Copilot-generated function to reverse a list using slicing or loop.  • Output: [3, 2, 1] for input [1, 2, 3]  • Observation on how adding a single example improved Copilot’s accuracy.  **List Reversal Functions (Earlier)**   * **Two implementations:**   + [**reverse\_list\_slicing(lst)**](vscode-file://vscode-app/c:/Users/prade/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)**- Uses Python slicing [::-1]**   + [**reverse\_list\_loop(lst)**](vscode-file://vscode-app/c:/Users/prade/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-browser/workbench/workbench.html)**- Uses backward loop with range** * **Example: Input [1, 2, 3] → Output [3, 2, 1]**   **Task 3: Few-shot Prompt – String Pattern Matching**  **Task Description #3**  • Write a comment with 2–3 examples to help Copilot understand how to check if a string starts with a capital letter and ends with a period.  **Expected Output #3**  • A function is\_valid() that checks the pattern.  • Output: True or False based on input.  • Students reflect on how multiple examples guide Copilot to generate more accurate code.  **Email Validation - Pattern Checking (Earlier)**   * **Function**: is\_valid(pattern, text) * **Purpose**: Validates text against regex patterns * **Demonstrated with 3 patterns**:   + Email pattern validation   + Phone number validation (XXX-XXX-XXXX format)   + Alphanumeric validation * **Returns**: True or False based on pattern match   **Task 4: Zero-shot vs Few-shot – Email Validator**  Task Description #4  • First, prompt Copilot to write an email validation function using zero-shot (just the task in comment).  • Then, rewrite the prompt using few-shot examples.  **Expected Output #4**  • Compare both outputs:  Zero-shot may result in basic or generic validation.  Few-shot gives detailed and specific logic (e.g., @ and domain checking).  • Submit both code versions and note how few-shot improves reliability.    **Zero-Shot vs Few-Shot Email Validator (Main comparison)**  **Zero-Shot Approach**   * **Function**: validate\_email\_zero\_shot(email) * **Logic**: Basic check - just looks for @ and . * **Accuracy**: ~44% (misses edge cases) * **Limitation**: Too generic, fails on invalid formats   **Few-Shot Approach**   * **Function**: validate\_email\_few\_shot(email) * **Logic**: Comprehensive validation:   + Exactly one @ symbol   + Non-empty username and domain   + No consecutive dots   + Domain must have at least one dot   + Extension must be ≥ 2 characters * **Accuracy**: ~100% (handles all cases) * **Advantage**: Multiple examples guide accurate code generation   **Task 5: Prompt Tuning – Summing Digits of a Number**  **Task Description #5**  • Experiment with 2 different prompt styles to generate a function that returns the sum of digits of a number.  Style 1: Generic task prompt  Style 2: Task + Input/Output example  **Expected Output #5**  • Two versions of the sum\_of\_digits() function.  • Example Output: sum\_of\_digits(123) ➝ 6  • Short analysis: which prompt produced cleaner or more optimized code and why?   1. **Trade-offs**: Understanding pros/cons helps choose the right implementation   **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** | | | | | | Week2 |  |