|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE** | | | | | **DEPARTMENT OF COMPUTER SCIENCE ENGINEERING** | | | | |
| **ProgramName:**B. Tech | | | | **Assignment Type: Lab** | | | **AcademicYear:**2025-2026 | | |
| **CourseCoordinatorName** | | | | Venkataramana Veeramsetty | | | | | |
| **Instructor(s)Name** | | | | 1.Dr. Mohammed Ali Shaik  2.Dr. T Sampath Kumar  3.Mr. S Naresh Kumar  4.Dr. V. Rajesh  5.Dr. Brij Kishore  6.Dr Pramoda Patro  7.Dr. Venkataramana  8. Dr. Ravi Chander  9. Dr. Jagjeeth Singh | | | | | |
| **CourseCode** | | | 24CS002PC215 | **CourseTitle** | | AI Assisted Coding | | | |
| **Year/Sem** | | | II/I | **Regulation** | | R24 | | | |
| **Date and Day**  **of Assignment** | | |  | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicableto**  **Batches** | |  | | | |
| **AssignmentNumber:3.3**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***ExpectedTime***  ***to complete*** |  |
|  | 1 | Lab 3: Prompt Engineering – Improving Prompts and Context Management    **Lab Objectives:**     * To understand how prompt structure and wording influence AI-generated code. * To explore how context (like comments and function names) helps AI generate relevant output. * To evaluate the quality and accuracy of code based on prompt clarity. * To develop effective prompting strategies for AI-assisted programming.       **Lab Outcomes (LOs):**  After completing this lab, students will be able to:     * Generate Python code using Google Gemini in Google Colab. * Analyze the effectiveness of code explanations and suggestions by Gemini. * Set up and use Cursor AI for AI-powered coding assistance. * Evaluate and refactor code using Cursor AI features. * Compare AI tool behavior and code quality across different platforms.           **Task Description#1**   * Try 3 different prompts to generate a factorial function.   **Expected Output#1**   * Comparison of AI-generated code styles   Prompt 1 : Generate a python function to calculate factorial of a number , use any number as example  CODE AND OUTPUT:  6d2e1d6fa6442692098e982cca058c08.png    PROMPT 2: Create a python function to find factorial of a number and use it in main function  CODE AND OUTPUT:  3754e75849c887ff852ea30f1377d53d.png    PROMPT 3: Generate a function in python that calculates factorial of a give number and take user input of the required number  CODE AND OUTPUT:  88cc791de096d05fa087ae2729cc159f.png    CONCLUSION :  All three codes calculate the factorial correctly, but the way they are written is different.   1. The first code uses a fixed number, so it’s just a simple example without user interaction. 2. The second code places the logic inside a main() function, which makes the program more organized and closer to real-world coding style. 3. The third code asks the user for input, making it interactive and flexible since you can test with any number.   In short, the difference is mainly in **structure** and **how the number is given**—example-based, organized with main(), or interactive with user input.      **Task Description#2**   * Provide a clear example input-output prompt to generate a sorting function.   **Expected Output#2**   * Functional sorting code from AI   PROMPT: Generate a sorting function in python that takes input from the user and provides appropriate output  CODE AND OUTPUT:  23a189149980d82256501a263aec36e7.png      **Task Description#3**   * Start with the vague prompt “Generate python code to calculate power bill” and improve it step-by-step   **Expected Output#3**   * Enhanced AI output with clearer prompts   PROMPT : Generate python code to calculate power bill  CODE AND OUTPUT:  23fe0533b3f531b75c1bb80ad49bc6af.png    Improvement : Generate python code to calculate power bill and to the power bill calculator include a fixed service charge of Rs.50 added to the total bill.  CODE AND OUPUT:  3483b3afb7b7698679d9db43f2fda012.png      **Task Description#4**   * Write structured comments to help AI generate two linked functions (e.g., login\_user() and register\_user()).   **Expected Output#4**   * Consistent functions with shared logic   PROMPT: generate a code that creates a login and register functions using user input and also add appropriate comments.  CODE AND OUTPUT:  c0e7463a0f93631321dcbe5d395046f1.png  9322fbdc13502297a3fab4131c4f47c4.png        **Task Description#5**   * Analyzing Prompt Specificity: Improving Temperature Conversion Function with Clear Instructions   **Expected Output#5**   * Code quality difference analysis for various prompts   PROMPT1 :Write a Python function to convert temperature.  CODE AND OUTPUT:  7d924aab55dabb7af77e007130da50ca.png    PROMPT2: Write a function to convert Celsius, Fahrenheit, and Kelvin temperatures  CODE AND OUTPUT:    e77c0ba80d926f11ee37b24ee118b2da.png  **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:** | | | | | | 03.08.2025 EOD |  |