**Name :Abhishekh Verma**

**Batch=06**

**Hall Tno.=2403A510H1**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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** | | | Week2-Tuesday | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicableto**  **Batches** | | 24CSBTB01 To 24CSBTB39 | | | |
| **AssignmentNumber:3.2**(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**   * Ask AI to write a function to calculate compound interest, starting with only the function name. Then add a docstring, then input-output example   Answer  A screenshot of a computer  AI-generated content may be incorrect.A screenshot of a computer  AI-generated content may be incorrect.  **Explaination:** **This program is a Compound Interest Calculator that:**   1. **Asks you for:**    * **How much money you’re starting with (principal)**    * **The yearly interest rate (%)**    * **How long you’ll keep it (years)**    * **How often interest is added (yearly, monthly, etc.)** 2. **Decides how many times a year the interest will be calculated (like 1 for yearly, 12 for monthly).** 3. **Uses the compound interest formula to figure out:**    * **The final amount after interest**    * **How much extra money you earned from interest** 4. **Shows the results nicely, with two decimal points.**   **💡 Example: If you put ₹10,000 at 5% yearly interest for 2 years, compounded monthly, the program tells you:**   * **Final amount after 2 years** * **Total interest earned**   **Expected Output#1**   * Comparison of AI-generated code styles   **Task Description#2**   * Do math stuff, then refine it to: # Write a function to calculate average, median, and mode of a list of numbers.     **Explaination: Simple Steps:**   1. **Get Input: Asks user to enter numbers separated by spaces** 2. **Calculate: Computes average, median, and mode** 3. **Display: Shows all three results**   **What Each Statistic Means:**   * **Average: Sum of all numbers divided by count** * **Median: Middle value when numbers are sorted** * **Mode: Number(s) that appear most frequently**   **Expected Output#2**   * AI-generated function evolves from unclear to accurate multi-statistical operation.   **Task Description#3**   * Provide multiple examples of input-output to the AI for convert\_to\_binary(num) function. Observe how AI uses few-shot prompting to generalize.   **imple Steps:**   1. **Get Input: Asks user to enter a decimal number** 2. **Convert: Uses the function to convert decimal to binary** 3. **Display: Shows the binary result**   **How the Function Works:**   * **Special case: If number is 0, returns "0"** * **Main logic: Repeatedly divides by 2 and collects remainders** * **Process:** * **Take remainder when divided by 2 (0 or 1)** * **Add it to the beginning of binary string** * **Divide number by 2 (integer division)** * **Repeat until number becomes 0**   **Expected Output#3**   * Enhanced AI output with clearer prompts   **Task Description#4**   * Create an user interface for an hotel to generate bill based on customer requirements   **Expected Output#4**   * Consistent functions with shared logic   **Simple Functions:**   1. **display\_menu(): Shows available services and prices** 2. **calculate\_room\_cost(): Calculates room charges ($100/night)** 3. **calculate\_food\_cost(): Calculates food charges ($25/meal)** 4. **calculate\_laundry\_cost(): Calculates laundry charges ($15/service)** 5. **calculate\_transport\_cost(): Calculates transport charges ($30/trip)** 6. **generate\_bill(): Creates and displays the final bill** 7. **main(): Controls the main program flow**   **How It Works:**   1. **Welcome: Displays welcome message** 2. **Menu Loop: Shows menu and gets user choice** 3. **Calculate Costs: User selects services and enters quantities** 4. **Generate Bill: Creates professional-looking bill with:**  * **Individual service costs** * **Subtotal** * **10% tax** * **Final total amount**  1. **Exit: Option to quit the program**   **Task Description#5**   * Analyzing Prompt Specificity: Improving Temperature Conversion Function with Clear Instructions   **Expected Output#5**   * Code quality difference analysis for various prompts * **Simple Functions:** * **celsius\_to\_fahrenheit(celsius): Converts °C to °F using (C × 9/5) + 32** * **celsius\_to\_kelvin(celsius): Converts °C to K using C + 273.15** * **fahrenheit\_to\_celsius(fahrenheit): Converts °F to °C using (F - 32) × 5/9** * **fahrenheit\_to\_kelvin(fahrenheit): Converts °F to K (via Celsius)** * **kelvin\_to\_celsius(kelvin): Converts K to °C using K - 273.15** * **kelvin\_to\_fahrenheit(kelvin): Converts K to °F (via Celsius)** * **display\_menu(): Shows available conversion options** * **main(): Controls the main program flow** * **How It Works:** * **Welcome: Displays welcome message** * **Menu Loop: Shows conversion options and gets user choice** * **Input: Asks for temperature value** * **Conversion: Performs the selected conversion** * **Result: Displays the converted temperature with proper units** * **Error Handling: Catches invalid inputs gracefully** * **Exit: Option to quit the progra**   **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** | | --- | --- | | Task#1 | 0.5 | | Task#2 | 0.5 | | Task #3 | 0.5 | | Task #4 | 0.5 | | Task #5 | 0.5 | | **Total** | **2.5 Marks** | | | | | | | 03.08.2025 EOD |  |