|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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** | | | | Venkataramana Veeramsetty | | | | | |
| **Instructor(s) Name** | | | | |  | | --- | | Dr. V. Venkataramana (Co-ordinator) | | Dr. T. Sampath Kumar | | Dr. Pramoda Patro | | Dr. Brij Kishor Tiwari | | Dr.J.Ravichander | | Dr. Mohammand Ali Shaik | | Dr. Anirodh Kumar | | Mr. S.Naresh Kumar | | Dr. RAJESH VELPULA | | Mr. Kundhan Kumar | | Ms. Ch.Rajitha | | Mr. M Prakash | | Mr. B.Raju | | Intern 1 (Dharma teja) | | Intern 2 (Sai Prasad) | | Intern 3 (Sowmya) | | NS2 ( Mounika) | | | | | | |
| **Course Code** | | | 24CS002PC215 | **Course Title** | | AI Assisted Coding | | | |
| **Year/Sem** | | | II/I | **Regulation** | | R24 | | | |
| **Date and Day**  **of Assignment** | | | Week2 - Monday | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicable to**  **Batches** | | 24CSBTB01 To 24CSBTB39 | | | |
| **Assignment Number:3.1**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***Expected Time to complete*** |  |
|  | **1** | **Lab Experiment: Prompt Engineering – Improving Prompts and Context Management (0.5 marks)**  **Objective**  To explore how prompt design and context influence AI-generated outputs and to learn techniques to improve AI responses.  Tools Required   * GitHub Copilot / Google Gemini / ChatGPT * VS Code / Google Colab * Internet access   Procedure   1. Select a simple task: *"Write a Python function to check if a number is prime."* 2. Use different prompting strategies to generate the solution: 3. Zero-Shot – no examples. 4. One-Shot – one example provided. 5. Few-Shot – multiple examples provided. 6. Context-Managed – detailed prompt with constraints and instructions. 7. Record AI responses and refine prompts to improve code quality. 8. Request AI to optimize the logic for efficiency. 9. Compare results and document improvements. 10. **Sample Prompts**  * Zero-Shot: Write a Python function to check if a number is prime. * One-Shot: Example: Input: 5 → Output: Prime. Now, write a function to check if a number is prime. * Few-Shot:   Example 1: Input: 7 → Output: Prime  Example 2: Input: 10 → Output: Not Prime  Example 3: Input: 2 → Output: Prime  Generate the function accordingly.   * Context-Managed (With Optimization)   Vs code/ github copilot:  A screenshot of a computer  AI-generated content may be incorrect.  **A screenshot of a computer program  AI-generated content may be incorrect.** | | | | | | Week2 - Monday |  |
|  | **2** | **A screenshot of a computer  AI-generated content may be incorrect.**  **Google gemini:**  **A screenshot of a computer  AI-generated content may be incorrect.**  **A screenshot of a computer  AI-generated content may be incorrect.**  **A screenshot of a computer  AI-generated content may be incorrect.**  **Chat gpt:**  **A screenshot of a computer  AI-generated content may be incorrect.**  **A screenshot of a computer  AI-generated content may be incorrect.**  **A screenshot of a computer  AI-generated content may be incorrect.**  **Google colab:**    **A screenshot of a computer  AI-generated content may be incorrect.**  **A screenshot of a computer  AI-generated content may be incorrect.**  **Task: Mobile Data Usage Billing Application (1.0 Marks)**  **Objective:**  Use Python programming and AI-assisted coding tools to create an application that simulates mobile data billing for a telecom service provider.  Instructions   1. Use GitHub Copilot or Google Gemini to assist in writing the program. 2. Read the following inputs from the user:    * Data Consumed (in GB)    * Plan Type (Prepaid / Postpaid)    * Additional Services Used (e.g., caller tune, OTT subscription, etc.) 3. Implement billing logic to calculate:    * DC (Data Charges) – charges based on data consumption    * VC (Value-added Charges) – charges for additional services    * Tax – applicable tax on the total bill 4. Display an itemized bill showing:    * Plan Type    * Data Usage and Charges    * Value-added Services and Charges    * Tax    * Total Bill Amount   Requirements   * Students must refer to their actual mobile bill for charge structure (data cost, service fees, taxes) to make the program realistic. * AI assistance (Copilot/Gemini) must be used to generate and refine the initial code.   Deliverables   * AI prompts used for code generation. * AI-generated Python code and any optimized version. * Screenshots of:   + AI interactions   + Program execution and output   + Comparison with the student’s actual mobile bill.   Copilot: | | | | | | Week2 - Monday |  |
|  | **3** | **A screenshot of a computer program  AI-generated content may be incorrect.**  **Output:**  **A computer screen shot of a black screen  AI-generated content may be incorrect.**  **Task: Develop an LPG Billing System (1.0 Marks)**  **Objective**  Apply your Python programming skills and utilize AI-assisted coding tools to build an application that calculates the LPG bill based on specified customer inputs and billing parameters.  Instructions   1. Use GitHub Copilot or Google Gemini to assist in writing and refining the program. 2. Read the following user inputs:    * Cylinder Type (Domestic 14.2 kg / Domestic 5 kg / Commercial 19 kg / Commercial 47.5 kg)    * Number of Cylinders Booked    * Subsidy Amount (applicable only for domestic cylinders) 3. Refer to the given LPG Price List to determine the price per cylinder:    * Domestic LPG (14.2 kg) → ₹905.00    * Domestic LPG (5 kg) → ₹335.50    * Commercial LPG (19 kg) → ₹1,886.50    * Commercial LPG (47.5 kg) → ₹4,712.00    * Delivery Charges (₹10 to ₹50) 4. Implement the billing formula:   Bill Amount = (Price per Cylinder × Quantity) - Subsidy (if applicable) + Delivery Charges   1. Calculate and display an itemized bill including:  * Cylinder Type * Number of Cylinders * Base Amount * Subsidy * Delivery Charges * Total Bill Amount   Deliverables   * A report containing:   + AI prompts used to generate the program   + AI-generated Python code   + Line-by-line explanation of the code   Copilot:  A screenshot of a computer program  AI-generated content may be incorrect.  A screenshot of a computer program  AI-generated content may be incorrect.  A screenshot of a computer  AI-generated content may be incorrect.  Output:  A screen shot of a computer  AI-generated content may be incorrect.  Explanation:  **Function: get\_cylinder\_price(cylinder\_type)**  python  CopyEdit  def get\_cylinder\_price(cylinder\_type):   * Defines a function to get the price based on cylinder type.   python  CopyEdit  prices = {  "Domestic 14.2 kg": 905.00,  "Domestic 5 kg": 335.50,  "Commercial 19 kg": 1886.50,  "Commercial 47.5 kg": 4712.00  }   * Dictionary storing prices for each type of cylinder.   python  CopyEdit  return prices.get(cylinder\_type, None)   * Returns the price for the given type, or None if invalid.   **Function: is\_domestic(cylinder\_type)**  python  CopyEdit  def is\_domestic(cylinder\_type):  return cylinder\_type.startswith("Domestic")   * Returns True if the cylinder type is domestic (starts with "Domestic").   **Main Program: main()**  python  CopyEdit  def main():  print("LPG Bill Calculator")   * Prints the program title.   python  CopyEdit  print("Cylinder Types:")  print("1. Domestic 14.2 kg")  print("2. Domestic 5 kg")  print("3. Commercial 19 kg")  print("4. Commercial 47.5 kg")   * Lists all available cylinder options.   python  CopyEdit  cylinder\_type = input("Enter Cylinder Type: ").strip()   * Takes user input for the cylinder type.   python  CopyEdit  num\_cylinders = int(input("Enter Number of Cylinders Booked: "))   * Takes input for the number of cylinders and converts it to an integer.   python  CopyEdit  subsidy = 0.0   * Initializes subsidy to 0.   python  CopyEdit  if is\_domestic(cylinder\_type):  subsidy = float(input("Enter Subsidy Amount (₹): "))   * If the type is domestic, ask for subsidy amount.   python  CopyEdit  else:  print("Subsidy not applicable for commercial cylinders.")   * Otherwise, print that commercial cylinders don’t get subsidy.   python  CopyEdit  delivery\_charges = float(input("Enter Delivery Charges (₹10 to ₹50): "))   * Get delivery charge as a float from user.   **Calculation Section**  python  CopyEdit  price\_per\_cylinder = get\_cylinder\_price(cylinder\_type)   * Get the price for the entered cylinder type.   python  CopyEdit  if price\_per\_cylinder is None:  print("Invalid Cylinder Type entered.")  return   * If the cylinder type is invalid, show error and exit the program.   python  CopyEdit  base\_amount = price\_per\_cylinder \* num\_cylinders   * Calculates base price without subsidy or delivery.   python  CopyEdit  total\_bill = base\_amount - subsidy + delivery\_charges   * Final bill after subtracting subsidy and adding delivery charges.   **Display the Final Bill**  python  CopyEdit  print("\n--- Itemized LPG Bill ---")   * Print header for bill section.   python  CopyEdit  print(f"Cylinder Type : {cylinder\_type}")  print(f"Number of Cylinders: {num\_cylinders}")  print(f"Base Amount : ₹{base\_amount:.2f}")  print(f"Subsidy : ₹{subsidy:.2f}")  print(f"Delivery Charges : ₹{delivery\_charges:.2f}")  print(f"Total Bill Amount : ₹{total\_bill:.2f}")   * Prints all the bill details in a formatted way.   **Run the program**  python  CopyEdit  if \_\_name\_\_ == "\_\_main\_\_":  main() | | | | | | Week2 - Monday |  |