PART-C

State the following based on your experience with LLMs for Requirements Engineering tasks:

1. How syntactically correct are LLM-generated user stories? Are they following INVEST framework? Provide examples from your problem specification.

Ans: LLM-generated user stories are generally syntactically correct but vary in clarity, conciseness, and structure. The Llama (LLM3) and Gemini (LLM4) models produce well-structured user stories that align with the INVEST (Independent, Negotiable, Valuable, Estimable, Small, Testable) framework. However, Deepseek (LLM1) and Mistral (LLM2) often generate lengthy or unclear statements that require manual refinement. Manually developed stories (Insem1) follow INVEST principles most effectively.

- Example:
  - Correct (Llama & Gemini)
    As a passenger, I want to buy a metro ticket so that I can travel easily.
    - Success: The system calculates fare, processes payment, and prints a ticket.
    - Failure: If payment fails, it displays an error message.
  - Incorrect (Mistral & Deepseek)
    As a user, I want the ticketing system to work effectively for all users.
    - Too not clearly definite, lacks specific steps or acceptance criteria.

- Conclusion:
  - Llama (LLM3) and Manual user stories are the most syntactically correct and follow the INVEST framework.
  - Deepseek (LLM1) and Mistral (LLM2) require improvements in structure and clarity.

2. How semantically correct are LLM-generated user stories?

Ans: Semantically, Llama (LLM3) and Gemini (LLM4) perform well, capturing the intent of user stories accurately. However, Deepseek (LLM1) and Mistral (LLM2) struggle with providing precise descriptions and sometimes generate generic or incomplete user stories that lack context. Manual user stories (Insem1) provide the highest level of clarity and completeness.

- Example:
  - Good Example (Llama & Gemini)
  - As a frequent traveler, I want automatic discounts so that I save money.
    - Success: The system detects travel history and applies the correct discount.
    - Failure: If discount criteria are not met, the system notifies the user.
  - Bad Example (Mistral & Deepseek)

As a user, I want discounts.

- Too general, missing conditions, lacks clarity.

- Conclusion:
  - Llama (LLM3) and Manual user stories are the most semantically accurate.
  - Deepseek (LLM1) and Mistral (LLM2) require human intervention to enhance clarity and detail.

3. Are LLMs capable of identifying the information about stakeholders and user stories from their own perspectives?

Ans: LLMs can identify some stakeholders, but they often miss key roles or generalize them. Llama (LLM3) and Gemini (LLM4) correctly recognize primary stakeholders, but they sometimes overlook secondary stakeholders like security teams and payment processors. Deepseek (LLM1) and Mistral (LLM2) often generate incomplete stakeholder lists. Manual user stories (Insem1) provide the most detailed and structured stakeholder identification.

- Example:
  - Good Stakeholder List (Llama & Gemini)
    - Passengers → Buy tickets, check balance, receive notifications.
    - Metro Staff → Handle customer queries, monitor system issues.
    - Security Team → Detect fraud, verify transactions.
    - Payment Gateway Providers → Process transactions securely.
  - Bad Stakeholder List (Mistral & Deepseek)
    - User, Metro Team, Government (Too broad, lacks specific roles).

- Conclusion:
  - Llama (LLM3) and Manual user stories provide the best stakeholder identification.
  - Deepseek (LLM1) and Mistral (LLM2) require refinement to ensure all relevant stakeholders are included.

4. Are LLMs capable of identifying the acceptance criteria (both success and failure) for the user story?

Ans: LLMs can define basic success and failure conditions, but they often lack detailed acceptance criteria.

- Llama (LLM3) and Gemini (LLM4) perform well, but sometimes miss technical failure cases.
- Deepseek (LLM1) and Mistral (LLM2) provide vague responses, often stating "System should notify of failure." without specifying conditions.
- Manual user stories (Insem1) define acceptance criteria most effectively, covering edge cases and specific failure scenarios.

  - Good Example (Llama & Manual)

        - Success: Ticket is printed, and SMS confirmation is sent.
        - Failure: If the card has insufficient funds, an error message is displayed.
    - Bad Example (Mistral & Deepseek)
        - Failure: "System should notify of failure." (Too brief, lacks details).

- Conclusion:
    - Llama (LLM3) and Manual provide the best acceptance criteria.
    - Deepseek (LLM1) and Mistral (LLM2) need improvement in defining clear failure cases.

Observation:

LLMs can help create user stories, but they still need human review to be clear, accurate, and complete. Some models, like Llama and Gemini, generate well-structured stories, but they may include extra details or miss key acceptance criteria. Others, like Deepseek and Mistral, often struggle with vague wording and missing important information. While LLMs save time, they can’t fully replace human thinking in understanding real-world needs. The best approach is to use AI for drafting and then refine the output manually to ensure it makes sense and meets all requirements properly.