

Day-5 Task Report – AI Question Generation Module

Name: Arya

Task: Optimize LLM prompts for better accuracy and consistency (MCQ Generation Pipeline)

Status: Completed

1. Work Summary

- Reviewed existing prompt performance and identified accuracy gaps such as weak distractors and JSON format breaks.
- Improved Bloom-level alignment to ensure distinct Recall, Application, and Analysis questions.
- Added negative constraints (e.g., “No ‘All of the above’”) and enabled strict JSON enforcement for backend stability.
- Strengthened distractor-generation logic to avoid obviously incorrect answers.
- Validated the improved Version-2 (V2) prompt using a real transcript chunk.

2. Issues Identified in Previous Prompt (V1)

Weak Distractors: Incorrect options were too obvious or irrelevant.

Inconsistent Taxonomy: Hard questions were often just simple recall questions.

JSON Instability: Occasional formatting errors caused backend parsing failures.

Repetitive Language: Question stems copied transcript text directly.

Vague Explanations: Explanations did not clearly justify correct vs. incorrect options.

3. Transcript Chunk Used for Optimization Testing

Currently we store the data in the form of arrays. An array contains similar types of elements.

For example, 4, 2, 90, and 1 are integer type elements. If I want to store floating values, then I must create a separate array where all the elements must be floating values.

An array stores the elements in a contiguous memory location. Array uses zero-based indexing,

which means the first element is stored at index 0. For example, the 4th element is stored at index 3.

4. Optimized Prompt (Version-2) – Final API-Ready Prompt

Role: You are an expert Psychometrician and Instructional Designer.

Output Requirement: Strict JSON only.

Task: Generate exactly 4 MCQs from the transcript chunk.

Difficulty Distribution (Bloom's Taxonomy):

- Question 1 – Easy (Recall): Direct fact or definition from text
- Question 2 – Medium (Application): Apply a concept to a scenario
- Question 3 – Medium (Application): Classify or interpret information
- Question 4 – Hard (Analysis): Infer structural or functional implications

Design Constraints:

- Exactly 4 options per MCQ
- Distractors must be plausible but incorrect
- Do not use “All of the above” or “None of the above”
- Avoid repeating transcript phrasing in the question stem
- Include correct_option_index for backend verification
- Output must strictly match the JSON schema

JSON Output Schema

```
{  
  "assessment_metadata": {  
    "topic_summary": "",  
    "blooms_distribution": "1 Recall, 2 Application, 1 Analysis"  
},  
  "questions": [  
    {  
      "id": 1,  
      "difficulty": "",  

```

```
        "question_stem": "",  
        "options": ["", "", "", ""],  
        "correct_option_index": 0,  
        "correct_answer_text": "",  
        "explanation": ""  
    }  
]  
}
```

5. Sample Output (Using Optimized Prompt V2)

(Demonstrates improved Hard/Analysis reasoning and strict JSON compliance.)

```
{  
    "assessment_metadata": {  
        "topic_summary": "Array data structure characteristics and indexing logic",  
        "blooms_distribution": "1 Recall, 2 Application, 1 Analysis"  
    },  
    "questions": [  
        {  
            "id": 1,  
            "difficulty": "Easy",  
            "cognitive_level": "Recall",  
            "question_stem": "According to the text, how are elements physically organized within  
an array's storage?",  
            "options": [  
                "In random memory locations",  
                "In a contiguous memory location",  
                "In a linked list format",  
                "In separate floating-point files"  
            ]  
        }  
    ]  
}
```

```
],
  "correct_option_index": 1,
  "correct_answer_text": "In a contiguous memory location",
  "explanation": "The transcript clearly states that an array stores its elements in a contiguous memory location."
},
{
  "id": 2,
  "difficulty": "Medium",
  "cognitive_level": "Application",
  "question_stem": "If a developer needs to access the 7th element in an array, which index should they reference?",
  "options": ["Index 7", "Index 6", "Index 8", "Index 0"],
  "correct_option_index": 1,
  "correct_answer_text": "Index 6",
  "explanation": "Zero-based indexing means the 7th element is at index 6."
},
{
  "id": 3,
  "difficulty": "Medium",
  "cognitive_level": "Application",
  "question_stem": "A user wants to store the values 4, 10, and 9.5 together. What constraint applies?",
  "options": [
    "They can be stored if the array is large enough",
    "They can be mixed because they are all numbers",
    "They cannot be stored in the same array because 9.5 is a floating value",
    "They must be converted to text first"
  ],
}
```

```
"correct_option_index": 2,  
  "correct_answer_text": "They cannot be stored in the same array because 9.5 is a  
floating value",  
  "explanation": "Arrays store similar types of elements; integers and floats require  
separate arrays."  
,  
{  
  "id": 4,  
  "difficulty": "Hard",  
  "cognitive_level": "Analysis",  
  "question_stem": "Why does the requirement for 'similar types of elements' restrict how  
arrays handle data input?",  
  "options": [  
    "It prevents the use of zero-based indexing",  
    "It forces the creation of separate data structures for different data formats",  
    "It ensures all elements are stored at index 0",  
    "It automatically converts all inputs to integers"  
,  
  "correct_option_index": 1,  
  "correct_answer_text": "It forces the creation of separate data structures for different  
data formats",  
  "explanation": "Because arrays enforce a single data type, handling mixed values requires  
designing separate storage arrays."  
}  
]  
}
```

6. Before/After Comparison Summary

Feature	Before Optimization (V1)	After Optimization (V2)
Distractors	Weak, obvious, sometimes illogical	Plausible, concept-aligned, challenging
Difficulty Levels	Hard questions often simple recall	Hard questions require inference & analysis
JSON Output	Sometimes unstable and required manual fixing	Strict, clean, schema-compliant
Question Phrasing	Repeated transcript wording	Paraphrased to test understanding
Explanation Quality	Vague	Clear justification for correct answer

7. Day-5 Conclusion

The optimized V2 prompt resolves prior issues related to distractors, clarity, JSON structure, and cognitive level alignment.

V2 is now fully reliable for backend integration and ready for use in the MCQ generation pipeline.

Submitted by:

Arya – AI / LLM Content Generation