Day-4 Task Report – AI Question Generation Module

Name: Arya

Task: Develop MCQ generation using Gemini API (Prompt & JSON Schema Design)

Status: Completed

Work Summary

- Converted MCQ generation prompt into a strict, API-ready format
- Added Bloom's Taxonomy difficulty levels (Recall, Application, Analysis)
- Designed a strict JSON schema for backend validation and LLM output control
- Ensured all constraints: 4 options, difficulty mapping, cognitive level tagging
- Generated sample output to test schema readiness
- Deliverable is ready for backend integration

Final Prompt Template (API-Ready)

You are an expert Psychometrician and Instructional Designer designed to output strict JSON.

Your task is to generate 4 Multiple Choice Questions (MCQs) from the provided transcript chunk.

DIFFICULTY DISTRIBUTION RULE:

1. Question 1 (Easy): Recall (Bloom Level 1). Ask for definitions or facts directly stated.

2. Questions 2 & 3 (Medium): Application / Concept (Bloom Levels 2–3). Ask to interpret or classify.

3. Question 4 (Hard): Analysis (Bloom Level 4). Require inference or synthesis of ideas.

DESIGN CONSTRAINTS:

- 4 options per question.

- Distractors must be plausible but incorrect.

- No "All of the above", "None of the above", or combined answers.

- Correct answer must be provably correct from the text.

- Explanation must be concise.

- OUTPUT MUST BE VALID JSON ONLY.

OUTPUT JSON SCHEMA:

```
[
  {
    "difficulty": "",
    "cognitive_level": "",
    "question_stem": "",
    "options": ["", "", "", ""],
    "correct_answer_text": "",
    "explanation": ""
  }
]
```

Transcript Chunk:

{{CHUNK_TEXT}}

Final JSON Schema (For Backend Integration)

```
{
  "name": "mcq_assessment",
  "strict": true,
  "schema": {
    "type": "object",
    "properties": {
      "assessment_metadata": {
        "type": "object",
```

```
  "properties": {
    "topic_summary": {
      "type": "string",
      "description": "A 5-word summary of the transcript topic."
    },
    "blooms_distribution": {
      "type": "string",
      "const": "1 Recall, 2 Application, 1 Analysis"
    }
  },
  "required": ["topic_summary", "blooms_distribution"],
  "additionalProperties": false
},
"questions": {
  "type": "array",
  "description": "A list of exactly 4 MCQs.",
  "items": {
    "type": "object",
    "properties": {
      "id": { "type": "integer" },
      "difficulty": { "type": "string", "enum": ["Easy", "Medium", "Hard"] },
      "cognitive_level": { "type": "string", "enum": ["Recall", "Application", "Analysis"] },
      "question_stem": { "type": "string" },
      "options": {
        "type": "array",
        "items": { "type": "string" },
        "minItems": 4,
        "maxItems": 4
```

```
          },
          "correct_answer_text": { "type": "string" },
          "explanation": { "type": "string" }
        },
        "required": [
          "id",
          "difficulty",
          "cognitive_level",
          "question_stem",
          "options",
          "correct_answer_text",
          "explanation"
        ],
        "additionalProperties": false
      }
    }
   },
   "required": ["assessment_metadata", "questions"],
   "additionalProperties": false
  }
}
```

Sample Output (Generated via Gemini 3 Pro)

```
{
  "assessment_metadata": {
    "topic_summary": "Array structure and indexing",
    "blooms_distribution": "1 Recall, 2 Application, 1 Analysis"
  },
```

```
"questions": [
 {
  "id": 1,
  "difficulty": "Easy",
  "cognitive_level": "Recall",
  "question_stem": "...",
  "options": ["...", "...", "...", "..."],
  "correct_answer_text": "...",
  "explanation": "..."
 },
 {
  "id": 2,
  "difficulty": "Medium",
  "cognitive_level": "Application",
  "question_stem": "...",
  "options": ["...", "...", "...", "..."],
  "correct_answer_text": "...",
  "explanation": "..."
 },
 {
  "id": 3,
  "difficulty": "Medium",
  "cognitive_level": "Application",
  "question_stem": "...",
  "options": ["...", "...", "...", "..."],
  "correct_answer_text": "...",
  "explanation": "..."
 },
```

```
    {
      "id": 4,
      "difficulty": "Hard",
      "cognitive_level": "Analysis",
      "question_stem": "...",
      "options": ["...", "...", "...", "..."],
      "correct_answer_text": "...",
      "explanation": "..."
    }
  ]
}
```

Day-4 Conclusion

Final API-ready prompt, strict validation schema, and sample output completed.
Backend team can now integrate Gemini API with predictable, validated MCQ generation.

Submitted by

Arya – AI / LLM Content Generation