MXB2026-Sylhet-Fakibazz-Master-Moshai

Please test our beginner app. This is the **first stage** of the application and **not the final version**.

Click the link below and open it in **Microsoft Edge** or **Google Chrome**.
Then, click on the **three-dot (⁝) menu**, select **“Add to Home Screen”**, and tap **Install**.
The application will then be installed on your device.

Link: https://mastermoshai-millionxbangladesh-fakibazz.lovable.app

Prompt & Process Documentation

Project Name: Master-Moshai — The AI Learning Companion

1-Purpose of Prompt Documentation

This section documents how **AI was actually used** in the development of Master-Moshai—not just as an API call, but as a **thinking, reasoning, and decision-making component**.

Judges evaluate this section to verify:

- Authentic AI usage
- Prompt engineering skills
- Explainable reasoning
- Iterative development process

2-Ideation & Problem Framing Prompts

Prompt Example 1:

```
Identify major problems in the Bangladeshi education system that can
be solved using AI. Focus on scalability and personalization.
```

Why this prompt was used:

- To explore real, high-impact educational challenges
- To avoid building a generic AI chatbot

AI Output Influence:

- Highlighted lack of personalized feedback
- Emphasized student-specific learning gaps
- Directly led to the concept of an AI learning companion

Prompt Example 2:

```
How can AI act as a personalized mentor for SSC and HSC students
instead of just a content provider?
```

Outcome:

- Shifted solution focus from “content delivery” to **learning intelligence**
- Defined Master-Moshai’s mentor-like behavior

3-Architecture & System Design Prompts

Prompt Example:

```
Design a scalable AI-powered education system using Node.js, MongoDB,
and LLM APIs. Include personalization and feedback loops.
```

Why this mattered:

- Structured frontend–backend–AI separation
- Introduced the **continuous learning loop**
- Inspired **modular AI orchestration logic**

Prompt for Multi-Model AI Strategy:

```
How can multiple AI APIs (OpenAI, Gemini, DeepSeek, Grok) be
orchestrated in a single system for reliability?
```

Influence on Design:

- Fallback and task-based model selection
- Reduced dependency on a single AI provider
- Improved system robustness

4-Coding & Agent Workflow Prompts

Backend Development Prompt:

```
Generate a Node.js and Express.js backend structure for an AI-powered
quiz and feedback system.
```

Used for:

- API route planning
- Controller-service separation
- Clean architecture design

AI Prompt Generation Logic:

```
Create a dynamic prompt that evaluates a student's answer, identifies
mistakes, and provides actionable improvement advice.
```

Resulting Feature:

- Personalized feedback instead of correct/incorrect labels
- Reasoning-based explanations

5-Evaluation & Reasoning Prompts

Performance Analysis Prompt:

```
Analyze the student's recent answers and identify weak concepts.
Suggest a personalized improvement strategy.
```

Why this is critical:

- Enables concept-level diagnosis
- Drives adaptive learning paths
- Forms the intelligence core of Master-Moshai

RAG-Specific Prompt:

```
Using the provided syllabus content and past performance data,
generate a response aligned with the curriculum.
```

6-How Prompts Influenced System Outputs

Prompt Type	AI Reasoning	System Output
Ideation	Problem identification	Project concept
Architecture	System modeling	Modular design
Coding	Logical structure	Clean backend
Evaluation	Performance analysis	Personalized feedback

This demonstrates a **clear reasoning chain** from prompts to AI decisions to final system outputs.

7-Iterative Prompt Refinement Process

- Initial prompts were broad
- Outputs were manually evaluated
- Prompts refined to:
    - Reduce ambiguity
    - Increase explainability
    - Align with educational goals

Impact:

- Improved response quality
- Increased relevance for students
- Enhanced AI reliability

8-Lovable.ai Features Integrated with Prompts

- Rapid UI creation using **Tailwind CSS and TypeScript**
- Multi-AI API integration for **OpenAI, Gemini, DeepSeek, Grok**
- Real-time testing and debugging
- Modular backend development (Node.js + Express.js)
- Built-in MongoDB connectivity for secure storage

Effect: Prompts combined with Lovable.ai allowed **rapid prototyping, adaptive learning, and emotional support features** while maintaining a clean, scalable architecture.

9-Why This Prompt Strategy is Effective

- Prompts are **goal-driven** and curriculum-aligned
- AI decisions are **explainable and traceable**
- Iterative refinement ensures **reliability, relevance, and clarity**

Behind This Project: