Language Learning Chatbot using Gemini AI and MongoDB

# Overview

This project is a command-line based language learning chatbot powered by Google's Gemini models. The chatbot acts as a personalized tutor, interacting with users in the target language and tracking their mistakes using a MongoDB database for improvement insights.

# Features

* - Interactive language practice with a generative AI model.
* - Tracks user mistakes for feedback.
* - Stores corrections in a MongoDB database.
* - Supports dynamic configuration via .env file.

# Tech Stack

|  |  |
| --- | --- |
| Component | Technology |
| Backend Logic | Python |
| AI Model | Google Gemini (via google.generativeai) |
| Database | MongoDB Atlas |
| Environment | Jupyter Notebook (.ipynb) |
| Config | python-dotenv |

# Architecture

Architecture Diagram:

+------------------+  
| User Interface | <---> CLI via Jupyter Notebook  
+--------+---------+  
 |  
 v  
+--------+---------+  
| Language Tutor | <---> Google Gemini API (generativeai)  
| (Chatbot Logic) |  
+--------+---------+  
 |  
 v  
+--------+---------+  
| Mistake Analyzer | --> Tracks wrong answers using keywords  
+--------+---------+  
 |  
 v  
+--------+---------+  
| MongoDB Atlas DB | <---> Stores user\_input & correction  
+------------------+

# System Workflow

1. User Inputs: User provides known language, target language, and proficiency.
2. Model Setup: System lists available Gemini models, selects a supported one (e.g., gemini-1.5-pro).
3. Chat Loop:
4. - Receives user message.
5. - Sends it to Gemini model along with system prompt.
6. - Displays AI-generated response.
7. - Checks response for keywords like “wrong” to log mistakes.
8. Mistake Tracking: Stores mistakes in MongoDB (mistakes collection).
9. Summary Display: At the end, shows all recorded mistakes to help the user learn.

# Environment Variables (.env)

GOOGLE\_API\_KEY=your\_google\_api\_key

MONGO\_URI=your\_mongodb\_connection\_string

# Conclusion

This project demonstrates how generative AI can be integrated with database systems for an interactive, feedback-driven learning experience. It combines cloud-based AI models and persistent storage to build a lightweight but scalable tutor bot.