

VECHO AI

December 17 2025

Chitrika, Yash , Maharshi

1. Introduction

Vecho AI is a context-aware conversational AI assistant developed using the Google Gemini API with the objective of enhancing student learning productivity. The system is designed to assist students in understanding academic concepts, completing assignments, generating summaries, and receiving instant answers through a conversational interface. Unlike traditional search engines, Vecho AI provides personalized and contextual responses, making learning more efficient and less time-consuming.

2. Goal of the Project

The primary goal of the Vecho AI project is to develop a smart conversational assistant that supports students in their academic journey. The system aims to:

- Answer academic and general knowledge questions accurately
- Explain complex topics in simple and student-friendly language
- Assist students in completing assignments and understanding problem statements
- Generate summaries, ideas, and explanations for better concept clarity
- Improve overall learning efficiency by reducing dependency on multiple platforms

b) Singleton Design Pattern

The Singleton pattern is applied to the Gemini API client. The API client is initialized only once and reused throughout the application lifecycle. This approach:

- Prevents repeated initialization of the API client
- Reduces resource consumption
- Ensures consistent API configuration

c) Strategy Design Pattern

The Strategy pattern is implemented to support different response modes based on user preference. Vecho AI provides multiple interaction strategies:

- Explanation Mode: Provides detailed and simplified explanations
- Summary Mode: Generates concise summaries of topics or content
- Q&A Mode: Offers direct and precise answers to questions

This pattern allows dynamic switching between response strategies without modifying the core logic.

c) Strategy Design Pattern

The Strategy pattern is implemented to support different response modes based on user preference. Vecho AI provides multiple interaction strategies:

- Explanation Mode: Provides detailed and simplified explanations
- Summary Mode: Generates concise summaries of topics or content
- Q&A Mode: Offers direct and precise answers to questions

This pattern allows dynamic switching between response strategies without modifying the core logic.

6. Technology Stack Used

Frontend

- HTML
- CSS
- JavaScript (Chat User Interface)

Backend

- Python
- Flask (Backend framework)
- Node.js (for supporting backend logic if required)

Generative AI

- Google Gemini API
- Prompt Engineering techniques for improved responses

Database

- SQLite
 - Used for storing chat history and interaction data
 - Enables context-aware conversations and learning continuity

Tools

- GitHub (Version control and collaboration)
- Visual Studio Code (Development environment)

7. Database Implementation

SQLite is used as the database system to store chat history, user messages, AI responses, and interaction modes. The database is integrated at the backend using Flask, ensuring secure and efficient data handling. Storing chat history allows Vecho AI to retrieve previous interactions and provide context-aware responses, enhancing the personalization of learning.

8. Team Structure and Roles

The project was developed collaboratively with clearly defined roles:

- Yash Bhati (58):
- Gemini API integration and backend development using Python and Node.js
- Chitrika Muthamma (23):
- Frontend development and user interface design
- Maharshi Goswami (18):
- JavaScript integration, database management, and project documentation

9. GitHub Repository Links

- Yash Bhati:

<https://github.com/yash08112/Gen-Ai-Project>

- Chitrika Muthamma:

<https://github.com/Chitrikamuthamma512/Gen-Ai-Project>

- Maharshi Goswami:

<https://github.com/MaharshiGoswami/agentic-ai>