Date	18 February 2026
Team ID	LTVIP2026TMIDS65753
Project Name	TransLingua – AI-Powered Multi-Language Translator
Maximum Marks	4 Marks

Solution Architecture Description

Solution architecture bridges the gap between business problems and technology implementation. In this project, the business problem is communication difficulty due to language barriers, and the technical solution is an AI-powered language translation system.

The TransLingua solution architecture is designed to provide a lightweight, secure, and scalable web-based application. It ensures structured communication between the user interface, application logic, and the Google Gemini Generative AI model.

The architecture achieves the following objectives:

- Provides an efficient technological solution to communication challenges.
- Clearly defines the structure and data flow within the system.
- Ensures secure API integration using environment-based key management.
- Supports scalability for future feature enhancements.

Architecture Components

The TransLingua system consists of the following main components:

1. User Interface (Streamlit)

- Accepts user input text.
- Allows selection of source and target languages.
- Displays translated output.

2. Application Logic (Python Backend)

- Validates user input.
- Constructs AI prompts dynamically.
- Manages API communication with Google Gemini.

3. AI Model (Google Gemini Generative AI)

- Receives structured translation prompt.

- Processes translation request.
- Generates translated output.

4. Security Layer

- API key stored securely in a `.env` file.
- Environment variables accessed using `python-dotenv`.
- `.env` file excluded from version control using `.gitignore`.

5. Output Module

- Displays translated results instantly on the web interface.

Architecture Flow

1. The user enters text and selects languages through the Streamlit interface.
2. The application logic processes the input and constructs a translation prompt.
3. The request is securely sent to Google Gemini Generative AI.
4. The AI model processes the request and generates translated output.
5. The translated text is returned and displayed to the user.

This architecture ensures efficiency, security, reliability, and scalability of the TransLingua application.

Solution Architecture for AI-Powered Multi-Language Translation Web Application

```
User              Streamlit          Python Application      Translated
(Web Browser) ──▶  User Interface ──▶ Logic             ──▶  Output

                   Environment Vairiabales
                   (.env security layer)
```