Requirements Analysis

Technology Stack

Date	**9 Feb 2026**
Team ID	**LTVIP2026TMIDS71831**
Project Name	**Translingua: ai-powered multi-language translator**
Maximum Marks	**2 Marks**

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2

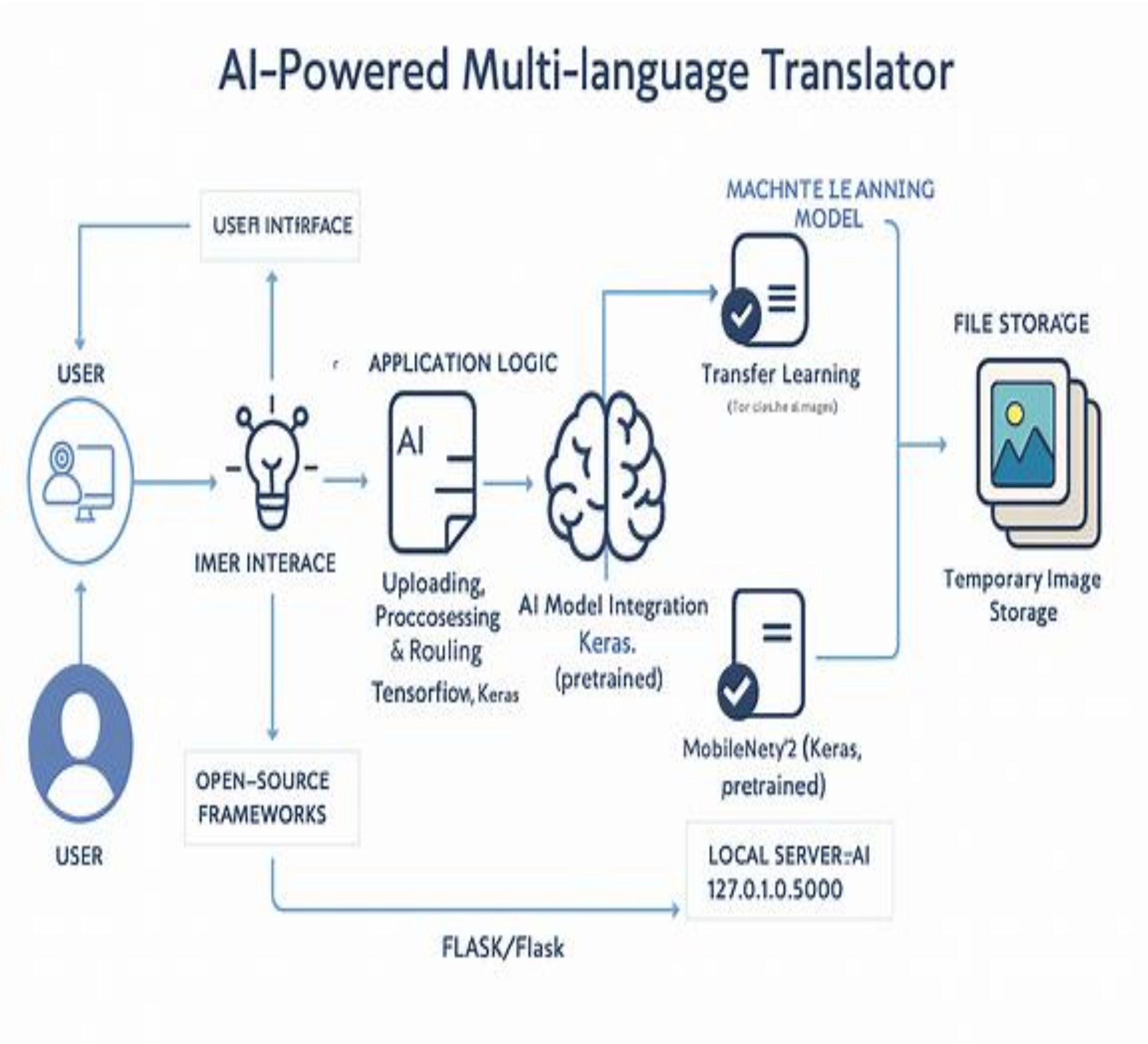

Table-1: Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	How users interact (text input, language selection, translation output)	HTML, CSS, Bootstrap 5
2	Application Logic-1	Handling user input, routing requests, and responses	Python, Flask
3	Application Logic-2	Language detection and translation inference	TensorFlow, Keras
4	Application Logic-3	Text preprocessing, tokenization, encoding & decoding	NLP Libraries, NumPy
5	Database	Stores translation history and user preferences	SQLite
6	Cloud Database	Optional cloud-based storage for scalability	Firebase / MongoDB Atlas
7	File Storage	Temporary storage for logs and translation cache	Local Filesystem
8	External API-1	Language detection service (optional hybrid approach)	Google Language Detect API
9	External API-2	Translation API (fallback or enhancement)	Google Translate API
10	Machine Learning Model	Neural Machine Translation for multi-language text	Transformer Model (Keras / TensorFlow)
11	Infrastructure	Runs on local system or cloud server	Localhost Flask / Cloud VM

Table-2: Application Characteristics:

S.No	Characteristic	Description	Technology Used
1	Open-Source Frameworks	Entire system built using open-source tools	Flask, TensorFlow, NumPy, Bootstrap
2	Security Implementations	Input validation, secure APIs, data privacy	Flask Security, HTTPS
3	Scalable Architecture	Modular architecture enabling easy language expansion	Flask MVC Architecture
4	Availability	Can be deployed on cloud for 24/7 access	AWS, Heroku, Docker-ready design
5	Performance	Real-time translation with low latency	Optimized Transformer Models