Signify

"बातचीत *Sabke Liye – Signify*"

Bridging Communication Through Technology

Signify

Team Members: Tannu, Maahi, Pintu

Guide: Mr. Ashish Aggarwal

Department: CSE, GIET Sonepat

Year: 2025

PROJECT INTRODUCTION

Abstract / Overview

- AI-powered communication assistant.
- Translates Indian Sign Language (ISL) to text and speech in real time.
- Combines Computer Vision, Deep Learning, and NLP.
- Promotes accessibility and inclusivity.

Problem Statement

- Communication barriers between hearing-impaired and non-signers.
- Few ISL translation tools available.
- Existing apps mostly support ASL.
- Need for a two-way real-time educational platform.

Introduction

- Signify is a **Sign Language to Text and Speech Conversion System.**
- Designed to improve communication for:
    1. People who are **deaf** or **hard of hearing.**
    2. People who **cannot speak.**
- Provides **real-time translation of sign language gestures.**
- Converts recognized gestures into **text output.**
- Uses a **camera** to **capture hand gestures.**

Objectives

- Eliminate communication barriers.
- Enable two-way digital interaction.
- Empower differently-abled users.
- Promote ISL learning through interactive tools.

Existing System

- **Depends on interpreters or written notes.**
- **Limited to one-way communication.**
- **Lacks ISL support and interactivity.**
- **No real-time or educational features.**

Proposed System

- **Provides real-time ISL translation.**
- **Supports sign-to-text, sign-to-speech, and text-to-sign.**
- **Includes dictionary, games, and quizzes.**
- **Uses AI and computer vision for accuracy.**

SYSTEM REQUIREMENTS

Software Requirements

- **Operating System:** Windows 10, Linux, or macOS
- **IDE/Editors:** Visual Studio Code, PyCharm, Jupyter Notebook
- **GUI–based Software:** Tkinter
- **Programming Language:** Python
- **DataBase:** LocalStorage (JSON/CSV)

Hardware Requiremnets

- **Processor:** Intel i5 or AMDRyzen 5 or higher.
- **RAM:**Minimum 8GB (16GB recommended for faster AI model processing).
- **Camera/Webcam:**HD camera for real-time hand gesture capture.
- **Devices:** Keyboard, Mouse, Monitor.

Functional Requirements

- **Sign-to-Text Translation:** Capture gestures and convert to text
- **Text-to-Sign Conversion:** Convert written text to ISL animation/video
- **Sign-to-Speech Conversion:** Speak out recognized signs
- **Dictionary Module:** Searchable ISL terms with meanings and animations
- **Quiz & Game Module:** Interactive learning environment
- **User Management:** Login, registration, and profile maintenance
- **Admin Control:** Manage content, signs, and quizzes

GUI Design and Importance

- Built using Tkinter (lightweight & easy to use)
- Event-driven design (responds instantly to user actions)
- Includes interactive components (buttons, frames, text boxes)
- Organized modules for Dictionary, Games, and Quizzes

System Architecture

- **Frontend:** Tkinter/React
- **Backend:** FastAPI, Node.js
- **ML Layer:** TensorFlow, MediaPipe
- **Database:** JSON/PostgreSQL

SYSTEM STUDY & DESIGN

Feasibility Study Overview

- Ensures success under cost, time, and tech constraints.

- Covers economic, technical, and behavioral feasibility.

Economic Feasibility

- Built using open-source technologies (Python, TensorFlow, OpenCV).
- Low hardware cost – only webcam and PC needed.
- High social and educational impact.

Technical Feasibility

- Uses Python, FastAPI, Tkinter, TensorFlow, MediaPipe.
- PostgreSQL for data management.
- Responsive UI with React and Tailwind.
- Scalable for mobile/AR platforms.

Behavioral Feasibility

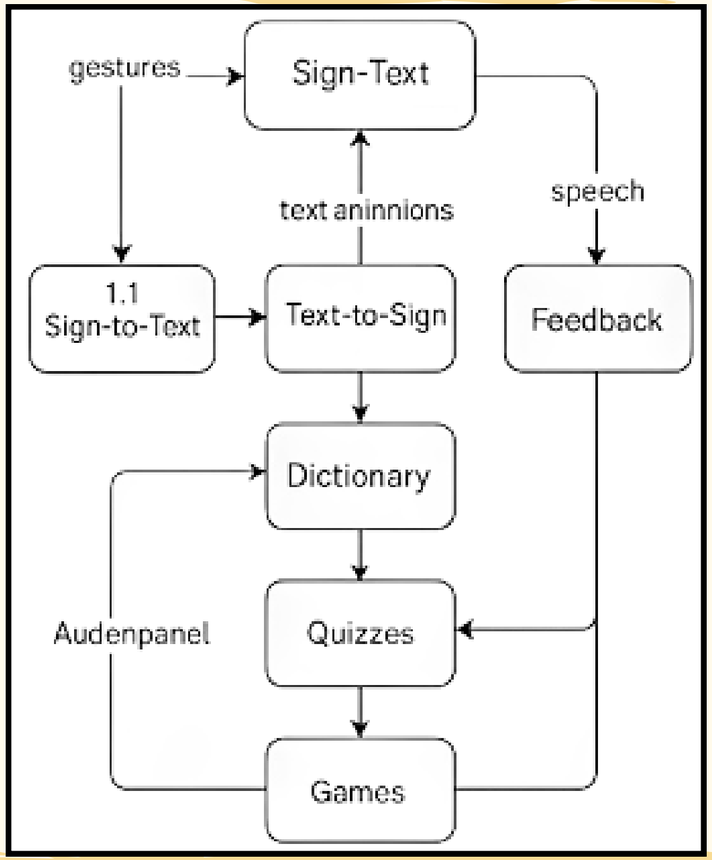
- Inclusive and intuitive interface.
- Gamified learning for engagement.
- Promotes awareness and inclusivity.
- High user adaptability.

Database Design

- Users, Translation History, Quiz tables.
- Ensures data integrity and personalization.

Data Flow Diagram

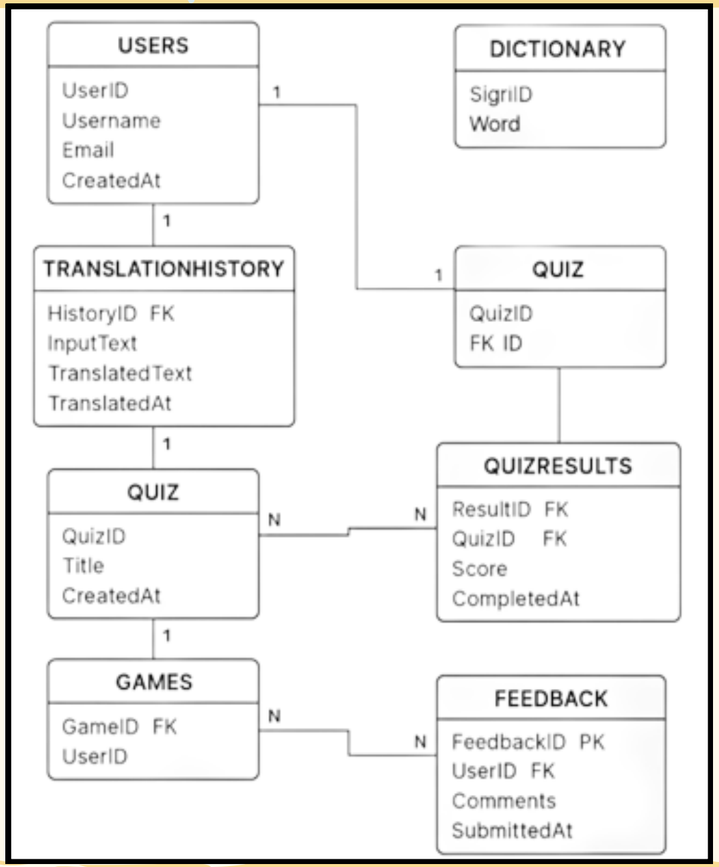
- Shows gesture-to-text processing.
- Illustrates logical flow of data.



E-R Diagram

- Entities: User, Dictionary, Quiz, History.

- Defines database relationships.



Implementation & Testing

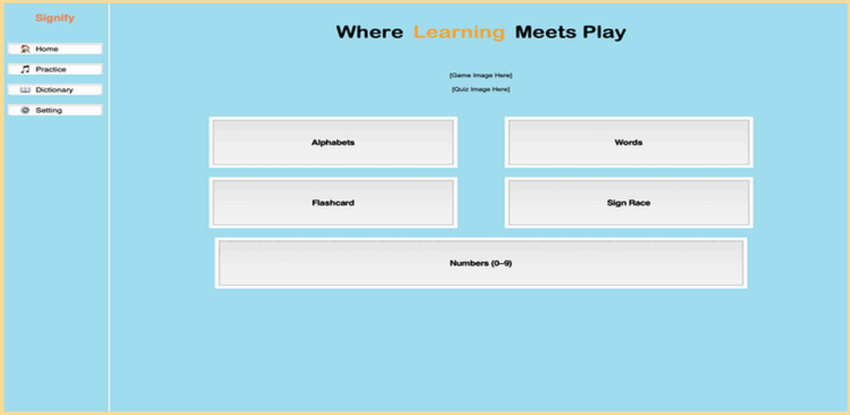
System Modules

- Login / Registration
- Sign-to-Text
- Text-to-Sign
- Sign-to-Speech
- Dictionary
- Quizzes & Games
- Admin Panel
- User Profile

System Screenshots

Starting Page

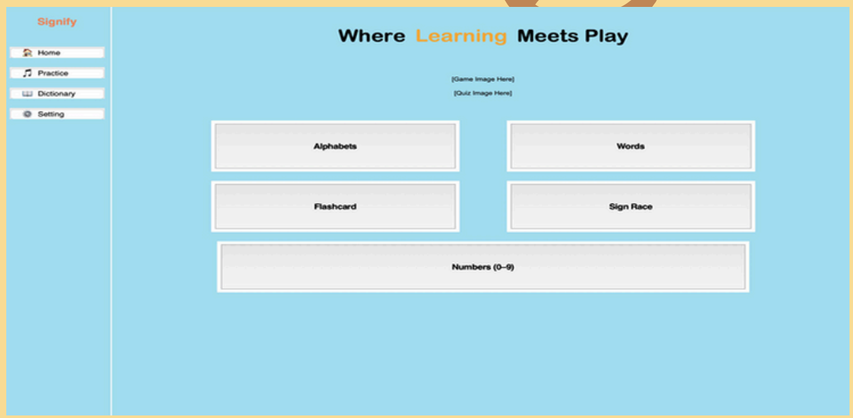
Main Dashboard



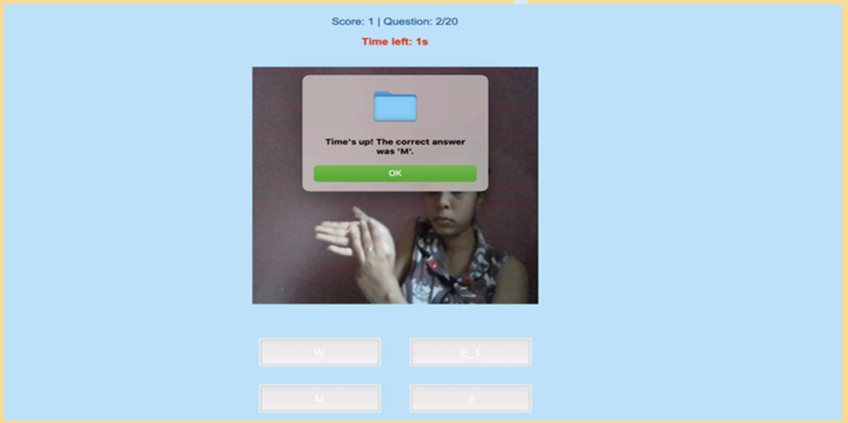
Practice Interface

Translation Interface

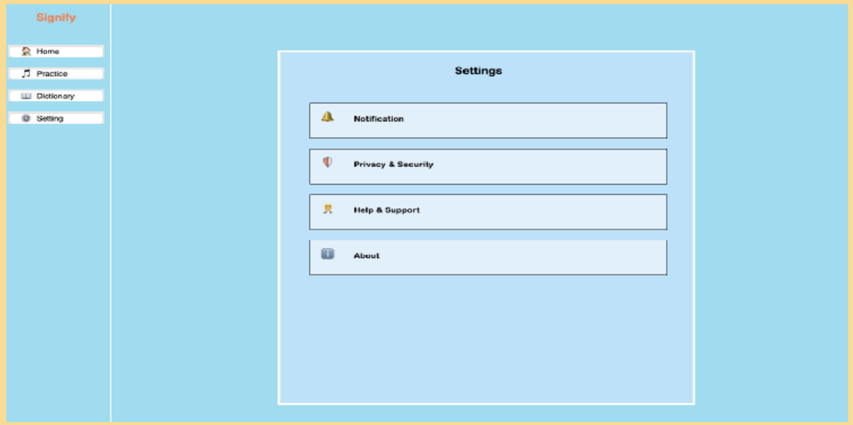
System Screenshots



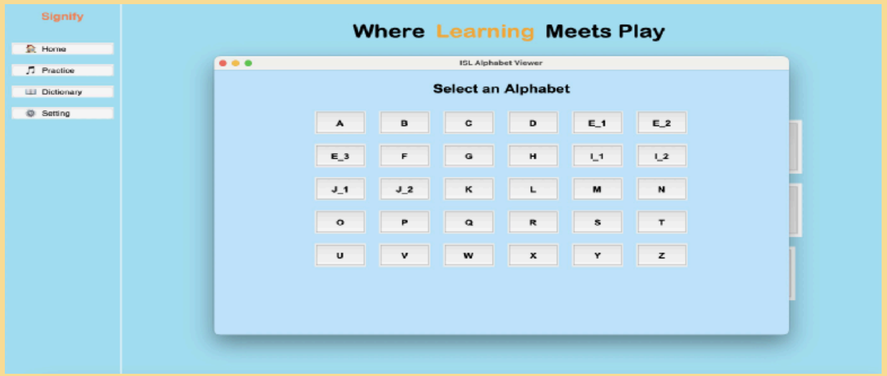
Dictionary Interface



Game Interface



Setting Interface



Alphabets Interface

Testing Overview

- Ensures quality, reliability, and correctness.
- Three levels of testing:
- Unit Testing
- Integration Testing
- System Testing

Test Case Design

- Check sign recognition accuracy.
- Validate text-to-sign and sign-to-speech output.
- Test dictionary search and quiz scoring.
- Ensure GUI responsiveness and error handling.

Types of Testing

- **White-box Testing:** Internal logic and code validation.
- **Black-box Testing:** User-level function and interface checks.
- Ensures both backend logic and frontend usability.

System Implementation

- Installation of hardware/software.
- Integration of ML models and datasets.
- System configuration and deployment.
- Verification and user approval.

Conversion Process

- Shift from manual to automated ISL communication.
- Integration of pre-trained ML models (.h5/.pkl).
- Data initialization for dictionaries and quizzes.
- Backup and recovery setup.

RESULTS & FUTURE SCOPE

Results and Discussion

- Successful real-time gesture recognition.
- High accuracy under various conditions.
- Smooth performance across modules.
- Positive user feedback from test groups.

Conclusion

- Signify bridges the communication gap for the deaf community.
- Promotes inclusivity, accessibility, and education.
- Demonstrates how AI can serve social good.

Future Scope

- Mobile app development.
- Integration with AR/VR for immersive ISL learning.
- Smart wearable compatibility.
- Multi-language and global sign support.