# Assistive Technology using Raspberry PI

Rebecca Nolan G00423921

Louise Deeth G0042

Sarah O’Connor G00423847

**Supervisor**: Daniel Cregg

**GitHub**: [/PPIT-Project](https://github.com/LouiseDeeth/PPIT-Project)

**Screencast**:

# Introduction

This project aims to develop a system capable of translating American Sign Language (ASL) gestures into text. Using a Raspberry Pi with an attached camera module, the system captures images of a user’s hand gestures. These images are processed using Google’s Gemini Pro Vision API, which interprets the gesture and returns the corresponding ASL meaning. A mobile-friendly frontend acts as the user interface, allowing remote control of the camera and real-time viewing of the translation.

# Requirements

* Capture high-quality images of hand gestures using the Pi camera
* Communicate with the Gemini API for ASL interpretation
* Develop a user-friendly frontend accessible from a mobile device
* Ensure secure handling of API keys
* Provide real-time feedback to the user
* Operate over a local network (Wi-Fi)

# Technologies Used

* **Hardware:** Raspberry Pi 5, Pi Camera Module v3
* **Backend:** Python, Flask, OpenCV, Google Generative AI (Gemini Pro Vision API)
* **Frontend:** React.js
* **Communication:** RESTful API over HTTP (JSON responses)
* **Version Control:** GitHub
* **Testing Tools:** Manual testing

# Design Methodologies

We adopted a modular design, separating the system into three main components:

1. **Frontend:** A React web app serving as the user interface.
2. **Backend:** A Flask server handling API requests, image capture, and interaction with Gemini.
3. **Camera Module:** Controlled via Python and OpenCV, dedicated to capturing images.

# Architecture of the solution

The system operates on a client-server model:

* **React Frontend (Client):**
  + Provides controls to the user (e.g., "Take Picture" button)
  + Sends requests to the Flask server
  + Receives and displays translated ASL text
* **Flask Backend (Server running on Raspberry Pi):**
  + Receives frontend requests
  + Captures an image using the Pi camera
  + Converts the image to Base64
  + Sends the image to Gemini Pro Vision API
  + Returns the API’s response to the frontend

This architecture ensures that the phone is used solely for interaction and display while the Pi handles processing and camera control.

Project management style

Limitations

Test Plans

Future Work

# Conclusion

The project successfully demonstrates a working prototype that can capture ASL gestures using a Raspberry Pi camera, process them via Gemini Pro Vision, and display the results on a mobile-friendly frontend. By leveraging hardware control, API integration, and modern web technologies, the system provides a practical solution for real-time ASL translation. Future improvements can enhance scalability, dynamic gesture recognition, and accessibility.