Anganwadi Centre Management System

Problem Statement: Smart Digital Solution for Anganwadi Centre Management Objective: Develop a scalable and user-centric software tool or mobile application to streamline operations, enhance transparency, and improve accountability in Anganwadi centres.

Smart Attendance Management System (SAMS)

**Vision and Purpose**

Our project aims to create an IoT-based, autonomous attendance and management system for Anganwadi Centres, focusing on accessibility and ease of use in rural and semi-urban areas. Recognizing the limited availability of mobile devices in these communities, our solution ensures that the operational complexity is minimized while maintaining efficiency and transparency.

By leveraging IoT technology and automating essential processes, we aim to provide a reliable, scalable, and user-friendly system that empowers Anganwadi workers and supervisors to manage operations seamlessly. This approach ensures that even in areas with constrained digital infrastructure, the system remains accessible, promoting inclusion and transparency.

**Why an IoT Approach?**

* Enhanced Accessibility: IoT devices like Raspberry Pi reduce the need for personal smartphones, ensuring that the system is accessible in regions with limited digital infrastructure.
* Offline Capability: The system is designed to operate effectively even with low or intermittent internet connectivity. Data synchronization is automated once a stable connection is available.
* Scalability and Cost-Effectiveness: The autonomous nature of the system, powered by simple yet effective hardware, ensures low operational costs and easy scalability to more centres.
* Minimal Training Required: With an intuitive, full-screen GUI, the system is designed for ease of use, ensuring that Anganwadi workers can operate it with minimal technical knowledge.

**Key Features**

1. Automated Facial Recognition for Attendance

Utilizes OpenCV and dlib libraries for real-time facial recognition.

Captures facial data through an AI camera module connected to a Raspberry Pi.

Automates attendance marking for both workers and children, eliminating manual entry errors.

Operates in offline mode, syncing data once connectivity is restored.

2. Sequential Process Automation

Designed to run multiple processes (Python scripts) sequentially with a single command.

Ensures the correct order of operations:

Capture Facial Features

Process Facial Data

Identify and Log Attendance

Display Attendance with Logs

A dedicated full-screen GUI allows users to either execute the entire process or skip to specific stages (like viewing attendance only).

3. Offline and Online Data Synchronization

Supports offline data storage for regions with low or intermittent internet connectivity.

Automatically syncs attendance data to MongoDB when a stable internet connection is available, ensuring data integrity.

4. Simplified Supervisor Reporting

Supervisors can log visits and attendance reports easily.

Initial implementation may leverage simple platforms like Google Forms, with the potential to integrate a dedicated reporting section in the future.

5. User-Friendly and Accessible Interface

Designed using CustomTkinter for a full-screen, distraction-free experience.

Offers intuitive one-click buttons for running scripts, minimizing operational complexity.

Includes real-time status updates for each process stage, ensuring clarity for the user.

**Technical Architecture**

Backend Framework:

Built using Python for core automation and data processing.

Flask framework used for web-based components (if integrated later).

Local Python scripts automate the attendance process and logging system.

Database Management:

SQLite allows the system to function autonomously, storing data locally and syncing with central servers when connected.

Frontend GUI:

Developed using CustomTkinter to provide a full-screen, interactive dashboard.

The dashboard displays process status, allows for sequential or specific script execution, and auto-handles web link operations.

Automation Process:

Utilizes Python's subprocess module to automate the execution of scripts.

Real-time status updates provide feedback on whether a process is running, completed, or encountered an error.

Internet Connectivity and Data Syncing:

Equipped with a 4G/GSM module for automatic data synchronization.

Ensures that even with limited internet availability, essential data is not lost.

**Hardware Components**

1. Raspberry Pi 5

Acts as the central processing unit for the system.

Supports the facial recognition scripts and handles local storage operations.

1. RPi AI Camera Module

Captures high-quality images for facial recognition.

Optimized for efficient processing even under low-light conditions.

1. Quectel EC200U-CN 4G/GSM Module

Provides internet connectivity for syncing attendance and reporting data.

Enables real-time data updates when the device is within network coverage.

1. Solar Power System (Optional for Remote Areas)

Consists of a solar panel, battery charge controller, and Li-ion battery.

Ensures uninterrupted operation in off-grid or power-deficient areas.

1. 2.8-inch Screen Module

Displays essential system status or prompts.

Can be integrated for user interaction if needed.

**Future Scope**

* Enhanced Supervisor Portal: Integrate supervisor reporting directly into the dashboard.
* Data Analytics Module: Introduce analytics for tracking attendance patterns and optimizing ration distribution.
* Mobile-Friendly Reporting: Develop a lightweight mobile interface for supervisors to report data remotely.
* Advanced Security Features: Implement enhanced data encryption for increased security in sensitive data handling.

**How to Setup the Program in Raspberry Pi:**

# After enabling camera module and updating all Raspberry Pi modifications

# Running Commands on Terminal

# Making the latest updates and upgrades for the setup

sudo apt-get update

sudo apt-get upgrade

# Downloading Python AI libraries and necessary packages

sudo apt install python3-venv

sudo apt-get install python3-numpy

sudo apt-get install python3-opencv

sudo apt-get install build-essential cmake

sudo apt-get install libgtk-3-dev

sudo apt-get install libboost-all-dev

# Download zip file for dlib Library https://drive.google.com/drive/folders/12It2jeNQOxwStBxtagL1vvIJokoz-DL4

extract zip file to SAMS folder

# Create a Virtual Environment

python3.11 -m venv --system-site-packages (named folder)

source (named folder)/bin/activate

# Installing Specific Libraries for Facial Recognition

pip install setuptools

pip install wheel

pip install dlib

pip install numpy

pip install scikit-image

pip install pandas

pip install opencv-python

pip install flask

# Installing the camera and UI packages

pip install picamera2

pip install customtkinter

# Now import packages to SAMS folder

# Hence the Program can run now