



PRESENTATION

By Team
Swathya Sahay

FEATURES INCORPORATED

1. Face ID Recognition for Smart Check-in

- Fast, contactless patient onboarding using facial authentication
- Modeled after Digi Yatri airport kiosks
- Patient consent-driven, privacy compliant (NDHM + eKYC)
- Linked to ABHA ID & PHR (health record) retrieval

2. LLM-Based Preliminary Diagnosis

- **Uses AI to assess symptoms entered by patient**
- **Provides a probable condition and suggests a care plan**
- **Doctor reviews it as a second opinion before proceeding**
- **Multilingual capability (e.g., Telugu, Hindi) for rural applicability**

ROADMAP: FROM CONCEPT TO NATIONWIDE DEPLOYMENT



Ideation

Identify gaps in rural healthcare and existing kiosks

Research

Study digital health initiatives (ABDM, eSanjeevani, NDHM, etc.)

MVP Design

Create user interface with face recognition, sensors



Prototype Development

Build offline-first Edge AI application



Pilot Launch

Deploy to 10 Ayushman Arogya Mandirs



Scale-Up

Partner with OEMs, states, telecom providers



Pilot Launch

Deploy to 10 Ayushman Arogya



Go-To-Market

Leverage CSR grants, align with health schemes

USES OF FEATURES AND MARKETING

Face Recognition:

Reduces wait time at OPD desks

Eliminates need to carry documents or ABHA card

Promotes digital, hygienic and automated health workflow

Elderly- and semi-literate-patient friendly

LLM-Based AI Diagnosis:

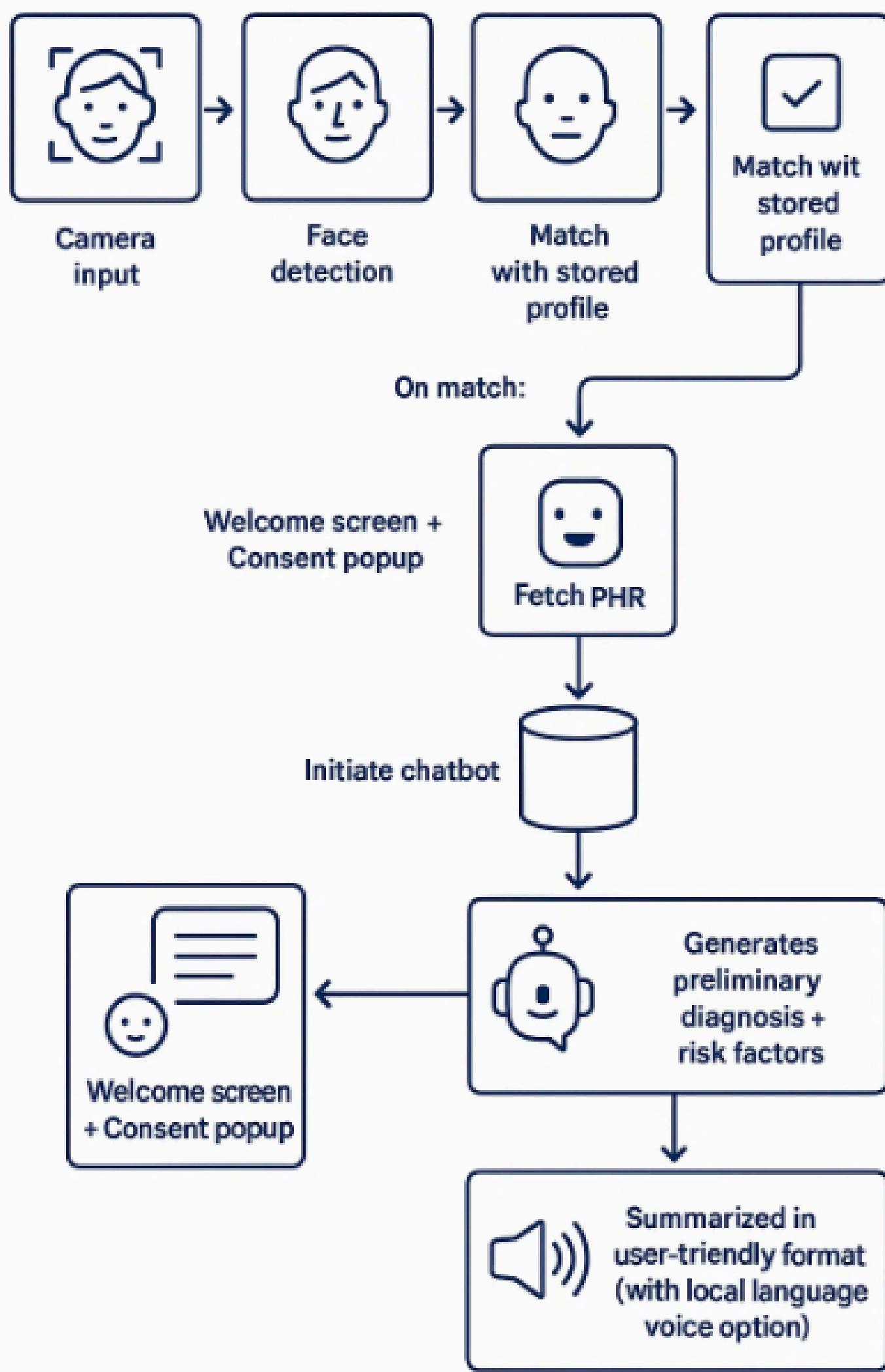
- Faster triage in high-footfall areas
- Reduces time burden on junior doctors
- Makes healthcare accessible in low-resource/rural zones
- Easy follow-up: AI suggests next step in care

Integration Flow & Architecture

PARTECHNICAL FLOW

- **CAMERA INPUT → FACE DETECTION → FACE EMBEDDING → MATCH WITH STORED PROFILE**
ON MATCH:
 - **→ WELCOME SCREEN + CONSENT POPUP → FETCH PHR → INITIATE CHATBOT**
 - **→ AI LISTENS TO SYMPTOMS → GENERATES PRELIMINARY DIAGNOSIS + RISK FACTORS**
 - **→ SUMMARIZED IN USER-FRIENDLY FORMAT (WITH LOCAL LANGUAGE VOICE OPTION)**

PARTECHNICAL FLOW



MODULES:

- **OPENCV + FACENET FOR FACE RECOGNITION**
- **LLM (GPT-4, LLAMA, ETC.) FOR AI RESPONSE**
- **NODE/FLASK + HTML UI ON KIOSK DEVICE**
- **INTEGRATION WITH INDIA STACK APIS: ABHA, DIGILOCKER
AGRAPH TEXT**

PROBLEM OBJECTIVE 1

"Can technologies like face ID recognition be brought in with advanced authorization from the patient, as seen in Digi Yatri Kiosks, to fasten the registration and check-in process?"

- Captures real-time face images from the webcam.
 - Uses Haar Cascade to detect faces in video frames.
 - Stores 30 grayscale face images per user with filenames like User.<id>. <count>.jpg.
 - Saves data in a folder named dataset/.
- Collects user metadata:**
- face ID (numeric), username, email
 - Logs data with timestamp and image count in an Excel file (user_data.xlsx) using openpyxl.
 - Ensures folder and file creation is handled safely.

Face Recognition Training Module

Features Developed:

- Uses LBPH (Local Binary Pattern Histogram) algorithm from OpenCV.
- Loads all images from the dataset/ directory.
- Automatically extracts Face ID from filenames.
- Detects and crops face regions using Haar Cascade.
- Trains the recognizer with all collected face data.
- Saves the trained model as trainer/trainer.yml.

Face Recognition Training Module

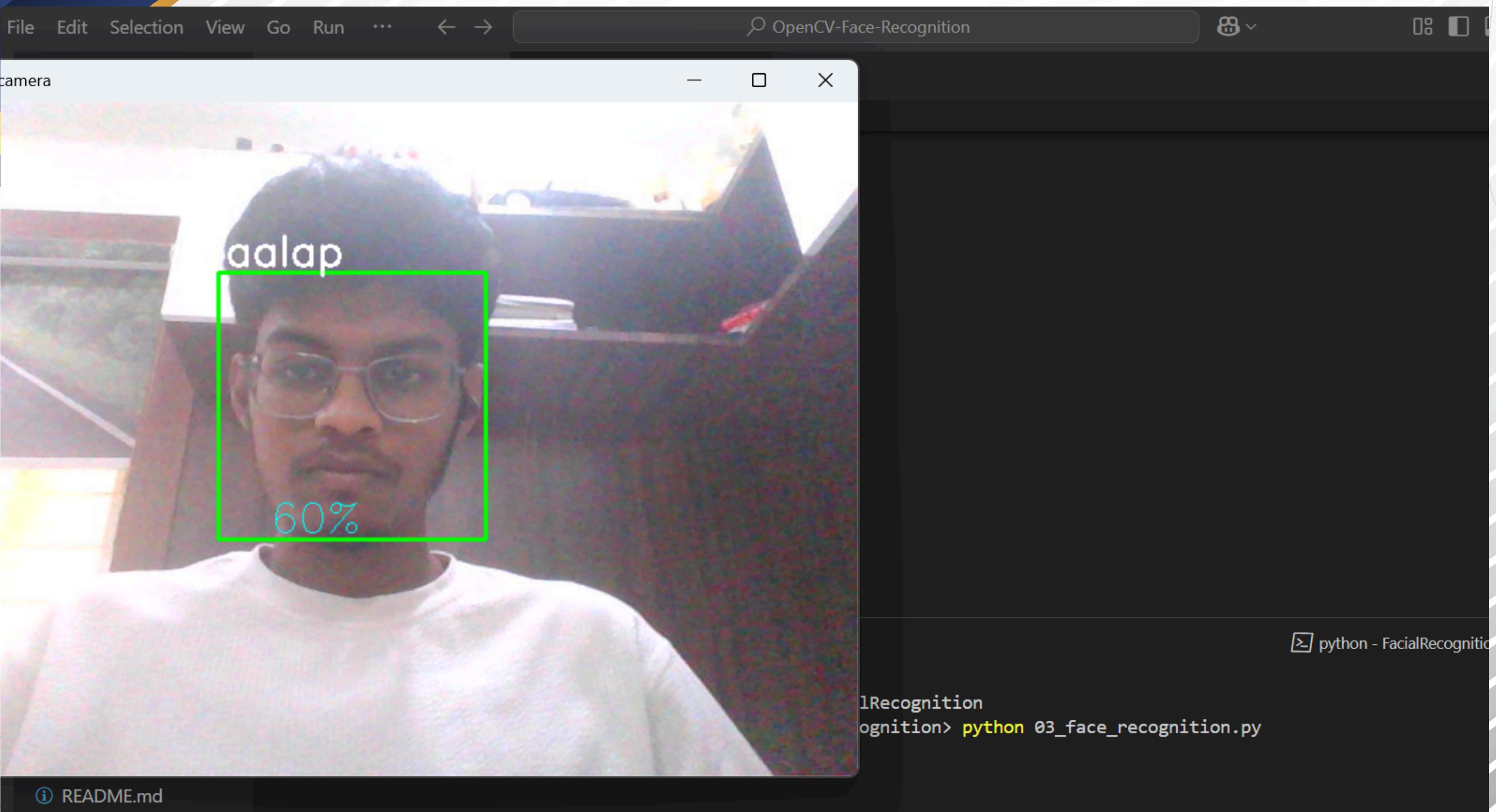
Features Developed:

- Uses LBPH (Local Binary Pattern Histogram) algorithm from OpenCV.
- Loads all images from the dataset/ directory.
- Automatically extracts Face ID from filenames.
- Detects and crops face regions using Haar Cascade.
- Trains the recognizer with all collected face data.
- Saves the trained model as trainer/trainer.yml.

Advantages

- Contactless, password-free authentication
- Multi-user support with unique face IDs
- Real-time recognition with live confidence scores
- Logged data for tracking users and login attempts

```
1
2 import os
3 import openpyxl
4 from datetime import datetime
5
6
7 cam = cv2.VideoCapture(0)
8 cam.set(3, 640)
9 cam.set(4, 480)
10
11 face_detector = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
12
13 face_id = input('\nEnter numeric user ID: ')
14 username = input('Enter username: ')
15 email = input('Enter email address: ')
16
17 print("\n[INFO] Initializing face capture. Look at the camera...")
18
19
20 if not os.path.exists('dataset'):
21     os.makedirs('dataset')
22
23
24 excel_file = 'user_data.xlsx'
25 if not os.path.exists(excel_file):
26     workbook = openpyxl.Workbook()
27     sheet = workbook.active
28     sheet.title = "User Data"
29     sheet.append(["Face ID", "Username", "Email", "Timestamp", "Image Count"])
```



DATA IN EXCEL SHEET

Siva Rama Krishna SR

user_data - Excel

Home Insert Draw Page Layout Formulas Data Review View Help Tell me what you want to do

Font Alignment Number Styles Cells Add-ins

Conditional Formatting Table Styles Cell Styles Add-ins

Sort & Filter Select Add-ins

Editing Add-ins

BACK UP THIS DOCUMENT Allow Copilot on this and other files backing using OneDrive (License required). Open OneDrive

X ✓ fx Face ID

Face ID	Username	Email	Timestamp	Image Count
01	aalap	adonga@f	2025-07-1	30

PROBLEM OBJECTIVE 2

"Can the Healthcare Kiosk conduct medical transcription for the case being discussed by using AI and LLMs (Large Language Models) that too for Indians in various Indian Languages and Dialects?"

- Accepts symptom input from patient in native language (e.g., Telugu, Hindi)
- Uses prompt-tuned LLM (like GPT or fine-tuned LLaMA) for symptom-to-condition mapping
- Handles both text and speech input from patients
- LLM analyzes common conditions, recommends initial actions (e.g., “check sugar levels”)
- Summarizes output in simple regional language for both patient and doctor
- Ensures patient consent, allows doctor override and validation

LLM-Based Diagnosis Pipeline

Features Developed:

- Utilizes pre-trained multilingual LLMs like GPT / LLaMA
- Accepts local language queries (via prompt templates or translation layer)
- Extracts symptoms and maps to disease clusters using healthcare ontology
- Generates probable diagnosis with suggested screening tests or next steps

Outputs in structured format:

- Diagnosis
- Confidence
- Triage Recommendation

Diagnosis Interface & Output Format

How It Works:

- Voice/text input → Transcription → Prompt preprocessor
- Symptom extraction & interpretation using custom health prompts
- LLM generates plain-language output with medical accuracy
- UI displays response in patient's selected language
- Output can be printed or sent to doctor dashboard for confirmation

Sample Output:

- The patient may have mild hypertension. Recommend checking BP thrice daily.”
- “This condition does not need emergency attention. Track symptoms for 3 days.”

Advantages

- Works for patients across multiple literacy levels
- Reduces initial diagnostic load on doctors
- Enables early screening before physical consultation
- Can be expanded to support 22 Indian languages
- Fully compatible with NDHM digital health records and ABHA flow

THANK YOU