INTEL INTERNSHIP REPORT

Project Title: **Product Category Creation for Healthcare Kiosks in India**

Submitted by: Aishwarya KV, Thanu shree B S

Branch: Electronics and Communication Engineering.

Institute: GITAM University Bengaluru.

Internship Duration: May – July.

Date of Submission: 11-07-2025.

**1.Abstract**

This project presents the design and implementation of a **Smart Healthcare Kiosk**, developed as part of the **Intel® Unnati** program, which aims to empower students with practical, industry-relevant skills. The kiosk demonstrates how self-service technology can automate routine healthcare tasks securely and efficiently.

The system integrates **biometric authentication** using face recognition with a **passkey-based security layer** to ensure only authorized users can access services. It simulates real-time **vitals measurement**, including heart rate, body temperature, and blood pressure, to mimic how actual medical sensors function in modern healthcare stations. Based on these vitals, a simple **decision-making logic** determines whether to dispense basic medication, showcasing how automated kiosks can support patient self-care and reduce manual workload for healthcare staff.

Developed using **Python**, the solution utilizes **OpenCV**, **dlib**, and **face\_recognition** libraries to handle real-time video capture, face detection, and face encoding. User data, including face embeddings and passkeys, are securely stored using **serialization**, modeling an **Electronic Health Record (EHR)** component in a simplified form.

This prototype reflects the vision of **Intel Unnati** to provide students with hands-on experience in building innovative solutions that address real-world problems. The kiosk demonstrates features such as **contactless check-in**, **vitals screening**, **conditional dispensing**, and **two-factor authentication**, all aimed at improving patient flow and making healthcare services more accessible.

The project can be further expanded with actual medical sensors, a physical medication dispensing unit, and cloud-based storage for maintaining patient health histories. Overall, it showcases a practical blend of **biometrics, security, and automation**, highlighting its potential use in hospitals, pharmacies, and remote clinics.

**2.Introduction**

In today’s fast-paced healthcare environment, there is an increasing need for systems that can automate routine tasks, improve efficiency, and reduce the burden on medical staff. One solution is the use of smart self-service kiosks, which have become common in hospitals, clinics, pharmacies, airports, and remote health centers. These kiosks help patients check in, monitor vital signs, access basic medical services, and even receive over-the-counter medication with minimal human intervention.

This project focuses on developing a prototype of a Smart Healthcare Kiosk that combines biometric face recognition, passkey-based user authentication, simulated vital signs monitoring, and conditional medication dispensing. The kiosk uses a webcam to capture the user’s face, which is processed using OpenCV, dlib, and the face\_recognition library in Python to generate a unique face encoding. This biometric data is compared to stored encodings to verify the user’s identity. To enhance security, the system implements a second layer of authentication using a user-defined passkey.

Once the user is authenticated, the kiosk simulates the measurement of vital health parameters such as heart rate, temperature, and blood pressure. Based on these vitals, the system makes simple decisions about whether to dispense medication, demonstrating the concept of a rule-based diagnostic support system.

This project showcases how technologies like computer vision, biometrics, and basic decision logic can be integrated to build a cost-effective self-service healthcare solution. It highlights the potential benefits of such kiosks in improving patient flow, providing 24/7 access to basic health services, and maintaining secure patient data. In the future, this prototype can be extended with real medical sensors, a physical dispensing unit, and cloud storage to create a fully functional automated healthcare kiosk.

## **3. Company Profile — Intel Corporation**

Founded in 1968 and headquartered in Santa Clara, California, **Intel Corporation** is a global leader in semiconductor manufacturing and a key innovator in computing and communications technology. Intel’s mission is to shape the future of technology, driving innovation to improve lives worldwide.

Its product portfolio includes microprocessors (such as the Intel® Core™ series and Xeon® processors), chipsets, System-on-Chips (SoCs), Field-Programmable Gate Arrays (FPGAs), memory and storage solutions (like Intel® Optane™), and networking components. Intel invests significantly in **Research and Development (R&D)**, advancing fields like **Artificial Intelligence (AI)**, **5G networks**, autonomous driving (through Mobileye), the **Internet of Things (IoT)**, and high-performance computing.

Additionally, **Intel Foundry Services (IFS)** extends Intel’s manufacturing capabilities to serve external clients globally, reinforcing its position as a trusted technology partner and semiconductor foundry.

Through initiatives like **Intel® Unnati**, the company collaborates with academic institutions to equip students with practical, industry-ready skills and resources, fostering the next generation of technology leaders.

## **4. Internship Objectives**

The main objective of this internship under the **Intel® Unnati Program** is to apply theoretical knowledge to a practical project that demonstrates the use of modern technology in healthcare.

The specific goals are:

* To develop a working **Healthcare Kiosk prototype** using **Python**, **OpenCV**, and **face recognition** libraries.
* To implement **biometric face authentication** combined with **passkey security** for two-factor user verification.
* To simulate **vital signs measurement** and demonstrate basic **decision logic** for medication dispensing.
* To gain hands-on experience with **secure data management** and **real-time video processing**.
* To build skills in troubleshooting, problem-solving, and applying **industry-relevant tools** to real-world applications.

## **5. Project Overview**

This project is a **Smart Healthcare Kiosk prototype** developed as part of the **Intel® Unnati Program** to demonstrate how technology can automate basic healthcare services in a secure and efficient way.

The system uses **biometric face recognition** with **passkey authentication** to verify a user’s identity through two-factor security. Once authenticated, the kiosk simulates the measurement of vital signs such as **heart rate, body temperature, and blood pressure**, mimicking how real medical sensors would work in an actual healthcare setting.

A simple **decision-making logic** uses these vitals to decide whether or not to dispense appropriate medication. For example, if the temperature is above a certain threshold, the system dispenses fever medication; otherwise, it may provide a general supplement.

This prototype showcases how contactless check-ins, secure identity verification, live health monitoring, and automated dispensing can be combined to make healthcare more accessible and reduce manual workload. It also highlights how such kiosks can be expanded in the future with real sensors, physical dispensing units, and cloud storage for patient health records.

This Smart Healthcare Kiosk prototype is implemented using **Python** as the primary programming language with a focus on **computer vision**, **biometric authentication**, and **decision-based automation**. The main components and their implementation are described below:

### **Face Recognition**

* The system uses a **webcam** to capture a live video feed.
* The **OpenCV** library is used for video capture and image processing.
* The **face\_recognition** library (built on top of **dlib**) generates a **128-dimensional face embedding** for each user.
* These embeddings are compared with stored data to recognize registered users with a configurable **tolerance threshold** to balance accuracy and false matches.

### **Passkey Authentication**

* After successful face recognition, each user must enter a **unique passkey** as a second layer of security (**two-factor authentication**).
* The passkey is created at the time of registration and securely stored along with the user’s face encoding.

### **Vitals Measurement**

* The kiosk simulates **real-time vitals** (heart rate, temperature, and blood pressure) using a Python function.
* Randomized or fixed values are generated to mimic a real sensor reading.
* This part demonstrates how real kiosks would integrate with actual **medical sensors**.

### **Medication Dispensing Logic**

* The system includes **simple decision logic** that checks the simulated vitals.
* Based on the conditions (e.g., high temperature), the kiosk “dispenses” medication by displaying which medicine would be provided.
* This models how **rule-based systems** support patient care in self-service kiosks.

### **User Data Storage**

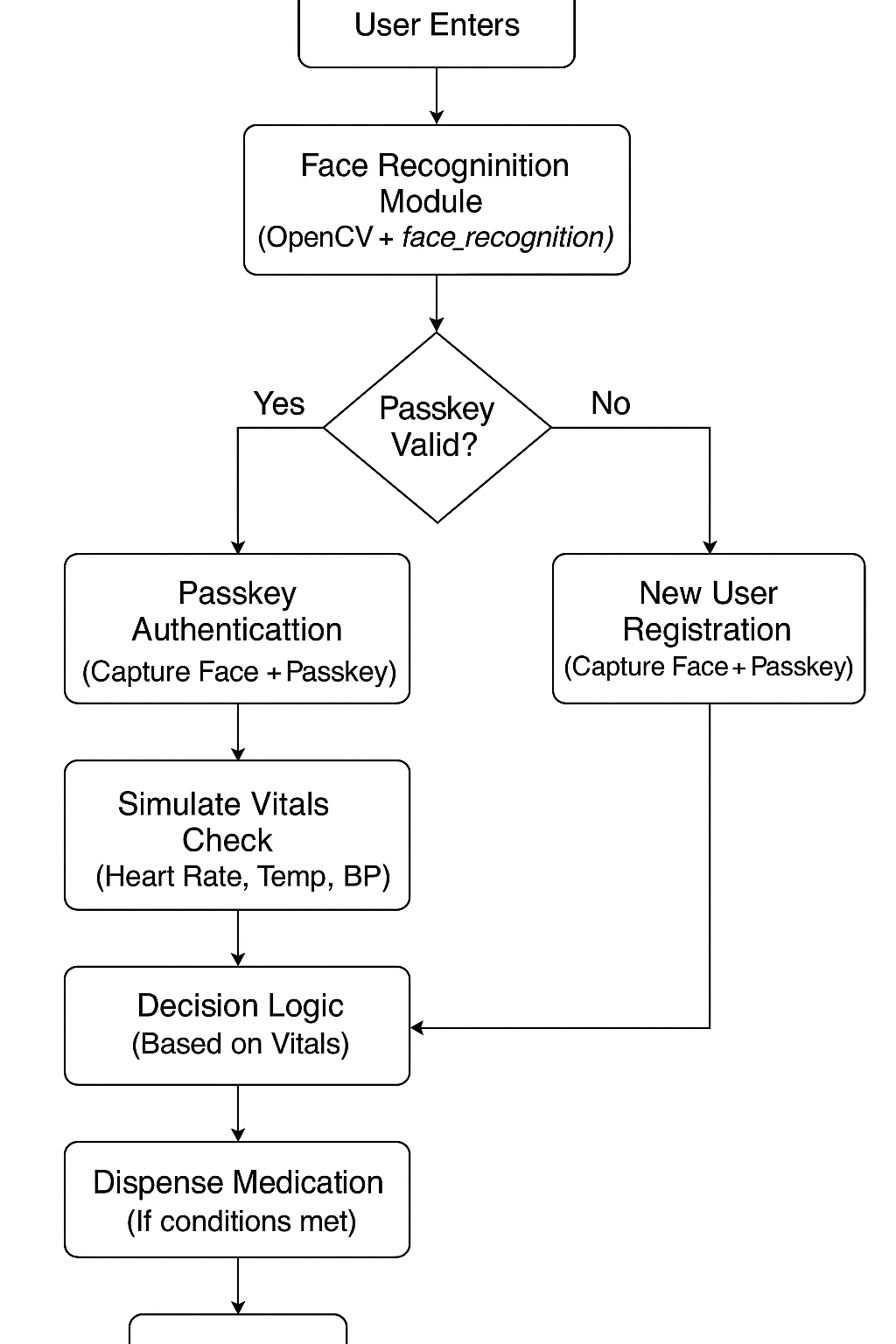
* User data, including **face encodings** and **passkeys**, is stored securely using **Python Pickle serialization**.
* The data is saved to a .pkl file, which acts like a lightweight database for this prototype.

### **Development Environment**

* A **Python virtual environment (venv)** was created to manage dependencies like OpenCV, dlib, cmake, and face\_recognition, ensuring compatibility and avoiding conflicts.

This end-to-end implementation demonstrates how **biometrics, security, live data simulation, and simple automation** can work together in a smart healthcare kiosk. The system can be expanded with real sensors, a physical dispensing unit, and cloud-based health record management in the future.

Block Diagram:



## **7. Module Description**

The Smart Healthcare Kiosk prototype consists of the following key modules:

### **1️ Face Recognition Module**

* Captures the user’s face using a webcam.
* Uses **OpenCV**, **dlib**, and **face\_recognition** libraries to detect and encode faces.
* Compares the live face encoding with stored data to recognize registered users.

### **2️ Passkey Authentication Module**

* Provides an extra layer of security using a user-created passkey.
* Verifies the passkey input for **two-factor authentication (2FA)** before granting access to kiosk services.

### **3️ Vitals Measurement Module**

* Simulates the measurement of vital signs: **heart rate**, **body temperature**, and **blood pressure**.
* Mimics how actual healthcare kiosks interface with medical sensors.

### **4️ Decision Logic Module**

* Applies simple rule-based conditions to check if medication should be dispensed.
* Uses vitals data as input to make automated decisions.

### **5️ Medication Dispensing Module**

* Demonstrates how the kiosk could dispense medication based on the decision logic.
* Currently prints the dispensing action to simulate the process.

### **6️ Data Management Module**

* Stores user data (face encoding and passkey) securely using **Python Pickle**.
* Maintains a simple **local database** for user records.

## **8. Results & Analysis (Current Status & Capabilities)**

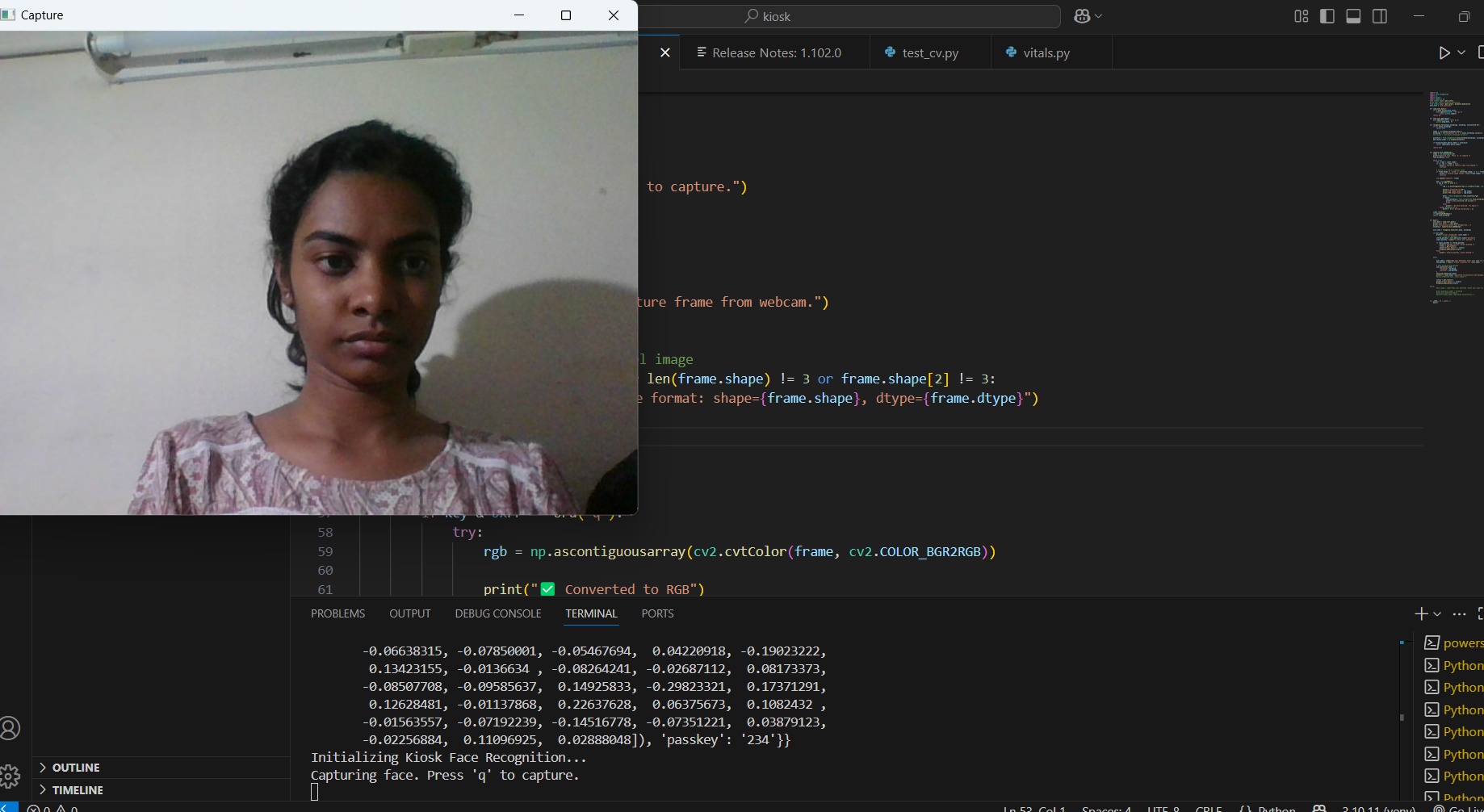
The **Smart Healthcare Kiosk prototype** has been successfully developed and tested to demonstrate its core functionalities. The current results and capabilities are as follows:

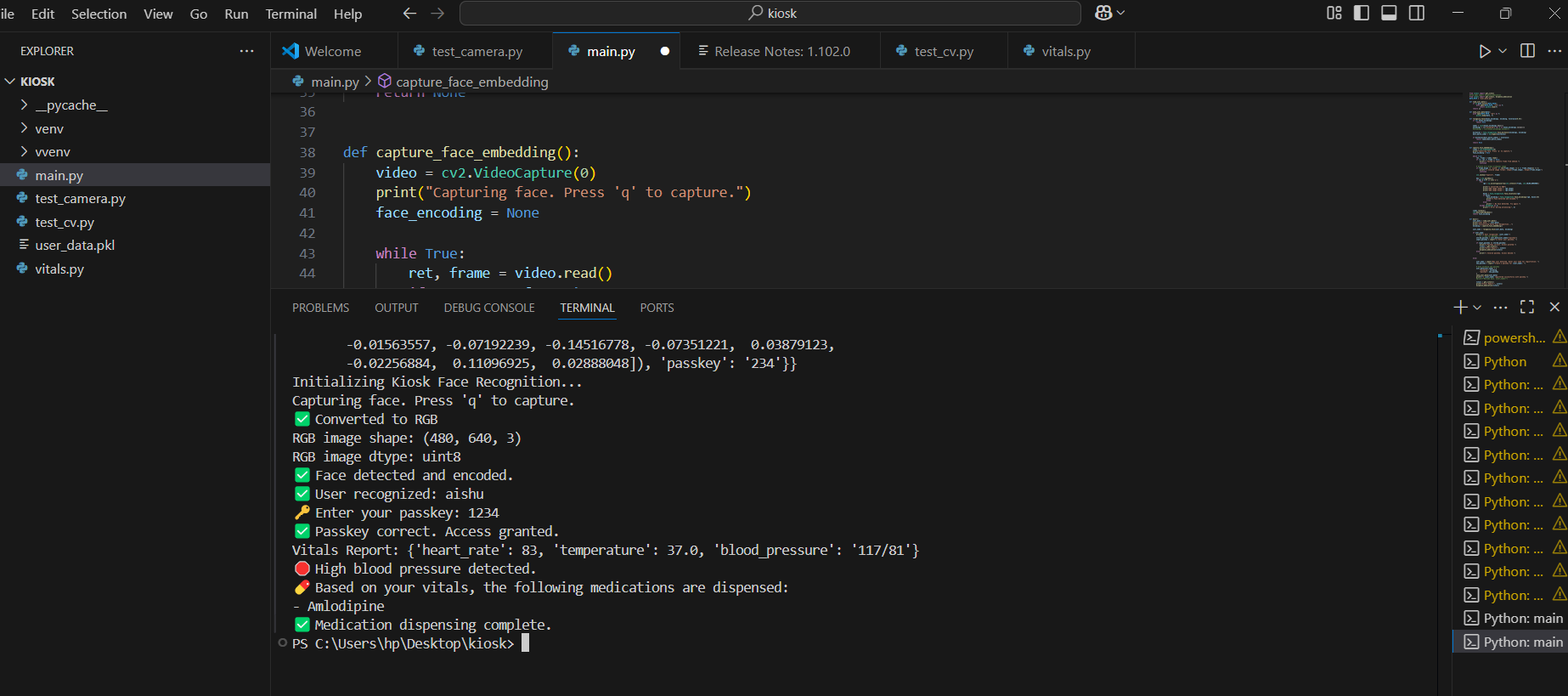
* **Face Recognition:** The system accurately captures and recognizes a registered user’s face using **OpenCV** and the **face\_recognition** library. It can differentiate between new and existing users with a configurable tolerance level for better accuracy.
* **Passkey Authentication:** The two-factor authentication works as intended. After face recognition, users must enter their unique passkey to gain access, adding an extra security layer.
* **Vitals Measurement:** The kiosk successfully simulates vital signs (heart rate, temperature, and blood pressure) using Python functions to mimic how real sensors would operate.
* **Decision Logic:** The program applies simple rule-based conditions to analyze vitals and decide whether medication should be dispensed. This demonstrates how basic diagnostic logic can work in real self-service kiosks.
* **Medication Dispensing:** While no physical dispenser is connected, the system displays which medication would be given based on the vitals check, proving that the logic works.
* **Data Management:** User data (face encodings and passkeys) are securely saved using Python Pickle, showing how simple **Electronic Health Records (EHR)** can be handled in a prototype.
* **Working Prototype:** The complete workflow — from capturing the face to passkey verification, vitals simulation, decision logic, and medication dispensing — runs smoothly, demonstrating the practical application of **biometrics**, **security**, and **healthcare automation**.

**Current Limitations**

* Real medical sensors and a physical dispensing unit are not yet integrated.
* Cloud storage and remote access features are not implemented in this version.

**Output:**





## **9. Challenges Faced**

During this project, a few challenges were faced:

* **Library Issues:** Installing **dlib**, **cmake**, and **face\_recognition** was tricky due to compatibility problems.
* **Face Recognition Accuracy:** Needed good lighting and fine-tuning the tolerance for reliable matches.
* **Passkey Integration:** Combining face recognition with passkey checks required careful logic.
* **Data Handling:** Saving and loading face data securely without errors took extra effort.
* **Simulating Vitals:** Mimicking real sensor values realistically needed clear logic

.

## **10. Key Learnings**

* Gained hands-on experience with **OpenCV**, **dlib**, and **face\_recognition** for real-time face recognition.
* Understood how to add an extra security layer using **passkeys** (2FA).
* Learned to create and manage a **Python virtual environment** to handle packages smoothly.
* Practiced storing user data securely using **Pickle serialization**.
* Improved my **debugging skills** while fixing installation and logic errors.
* Built a simple **decision logic system** to automate medication dispensing based on vitals.
* Realized how **biometrics**, sensors, and rule-based systems can make healthcare more efficient.

## **11. Conclusion**

In conclusion, this project demonstrates how a simple **Smart Healthcare Kiosk** can integrate **face recognition**, **passkey authentication**, and **vital signs simulation** to automate basic healthcare tasks securely.  
 Through this internship, I gained practical experience in using **OpenCV**, **Python**, and biometric authentication, and learned how simple decision logic can help in automating real-world healthcare services.  
 This prototype shows that such kiosks can reduce manual workload, improve patient self-service, and be further expanded with real sensors, cloud storage, and physical dispensing units in the future.

## 

## **12. Executive Summary**

This Healthcare Kiosk project was developed under the **Intel® Unnati Program** to demonstrate how modern technologies like **biometrics**, **computer vision**, and **automation** can improve healthcare delivery.  
 The system uses a webcam with **OpenCV** to capture and recognize user faces, combines it with passkey-based two-factor authentication, simulates vital sign measurements, and uses rule-based logic to decide on medication dispensing.  
 This prototype highlights the potential of contactless, secure, and automated healthcare solutions that can make medical services more accessible and efficient.  
 The internship also provided hands-on learning with **Python virtual environments**, library management, and practical problem-solving — bridging academic knowledge with real-world applications.

## **13. References**

1 **OpenCV Library**<https://opencv.org/>

2 **dlib Library**<http://dlib.net/>

3 **face\_recognition Library (by Adam Geitgey)**<https://github.com/ageitgey/face_recognition>

4 **Python Official Documentation**<https://docs.python.org/3/>

5 **CMake Official**<https://cmake.org/>

6 **Pickle Module (Python Docs)**<https://docs.python.org/3/library/pickle.html>

7 **NumPy Library**<https://numpy.org/>