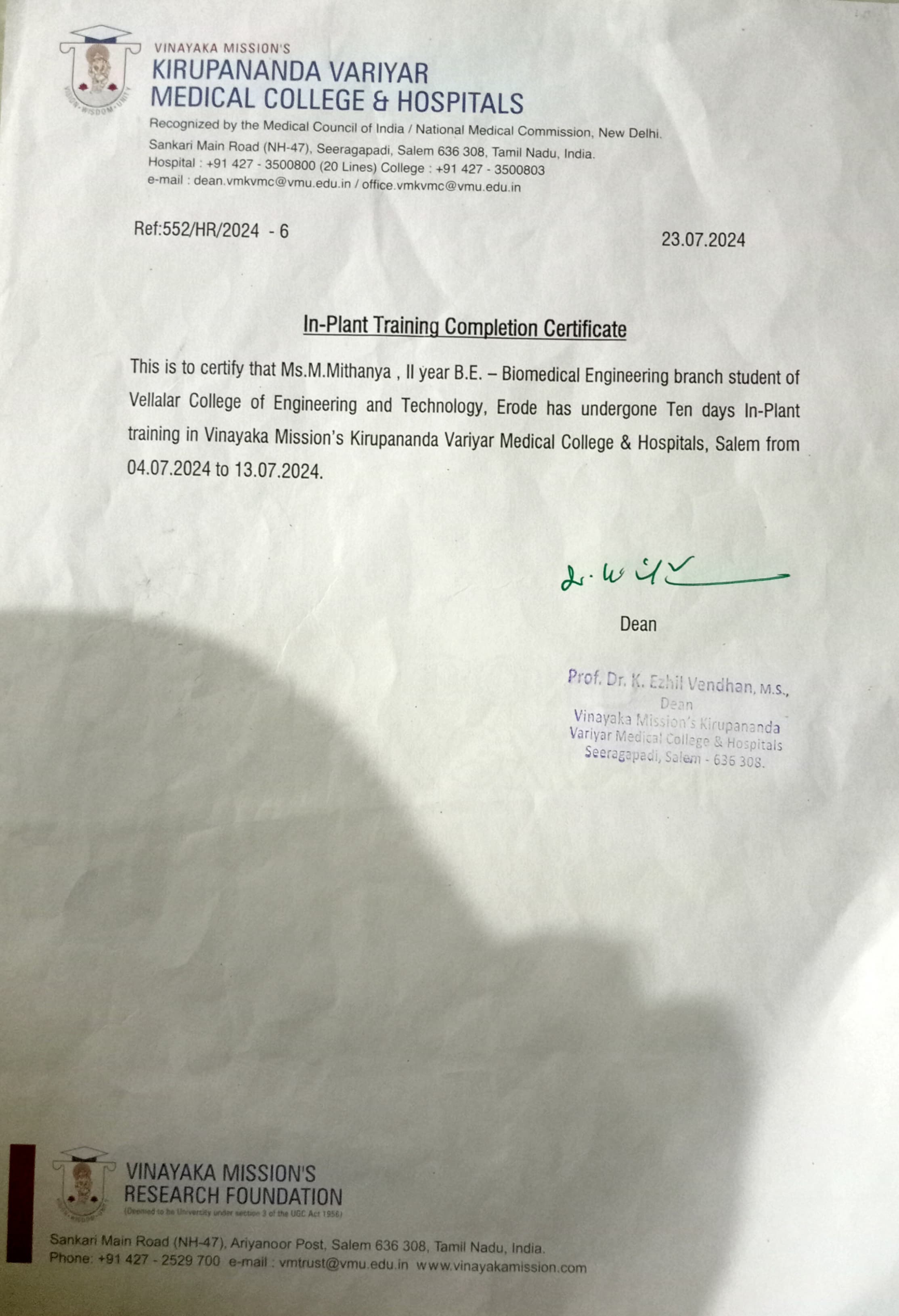
**10 days of exploring ICU systems, diagnostic devices, and medical gas pipelines – and translating them into real Python programs.**

**Here’s my journey where healthcare meets technology .**

****

**Day 1 Immersing in Biomedical Devices**

**Walking into the biomedical lab on Day 1 felt like stepping into the future of healthcare. I explored Defibrillators, Infusion Pumps, and Syringe Pumps, learning how these devices are vital in saving lives. Inspired, I built a Python mini system to admit, view, and discharge patients replicating what happens in a real hospital workflow.**

**PYTHON CODE:**

**patients = []**

**def admit\_patient():**

**name = input("Enter patient name: ")**

**age = input("Enter patient age: ")**

**issue = input("Enter health issue: ")**

**patients.append({"Name": name, "Age": age, "Issue": issue})**

**print(f"Patient {name} admitted successfully.\n")**

**def view\_patients():**

**if patients:**

**for p in patients:**

**print(f"{p['Name']} - {p['Age']} yrs - {p['Issue']}")**

**else:**

**print("No patients admitted.\n")**

**def discharge\_patient():**

**if patients:**

**name = input("Enter patient name to discharge: ")**

**for p in patients:**

**if p['Name'] == name:**

**patients.remove(p)**

**print(f"Patient {name} discharged successfully.\n")**

**return**

**print("Patient not found.\n")**

**else:**

**print("No patients to discharge.\n")**

**while True:**

**print("1. Admit Patient\n2. View Patients\n3. Discharge Patient\n4. Exit")**

**choice = input("Choose: ")**

**if choice == "1":**

**admit\_patient()**

**elif choice == "2":**

**view\_patients()**

**elif choice == "3":**

**discharge\_patient()**

**elif choice == "4":**

**break**

**else:**

**print("Invalid choice.\n")**

**OUTPUT:**

**Admit Patient**

**2. View Patients**

**3. Discharge Patient**

**4. Exit**

**Choose: 1**

**Enter patient name: Ravi**

**Enter patient age: 40**

**Enter health issue: Fever**

**Patient Ravi admitted successfully.**

**Day 2 – Understanding Medical Gas Systems**

**Ever wondered how Oxygen travels from tanks to patient beds? On Day 2, I understood how hospitals manage medical gases like Oxygen and Nitrogen. To translate this experience digitally, I created a Python program that lets wards place gas requests and track their supplies efficiently.**

**requests = []**

**def add\_request():**

**ward = input("Enter ward name: ")**

**gas = input("Enter gas type: ")**

**requests.append({"Ward": ward, "Gas": gas})**

**print(f"Request for {gas} added for {ward}.\n")**

**def view\_requests():**

**if requests:**

**for r in requests:**

**print(f"Ward: {r['Ward']} - Gas: {r['Gas']}")**

**else:**

**print("No pending requests.\n")**

**while True:**

**print("1. Add Request\n2. View Requests\n3. Exit")**

**choice = input("Choose: ")**

**if choice == "1":**

**add\_request()**

**elif choice == "2":**

**view\_requests()**

**elif choice == "3":**

**break**

**else:**

**print("Invalid choice.\n")**

**OUTPUT:**

**Add Request**

**2. View Requests**

**3. Exit**

**Choose: 1**

**Enter ward name: ICU**

**Enter gas type: Oxygen**

**Request for Oxygen added for ICU.**

**Day 3 – Simulating OT Scheduling**

**The Operation Theatre (OT) taught me about time-critical surgeries and scheduling. Using this learning, I developed a Python tool that helps schedule surgeries and view upcoming operations, bringing OT workflows to code.**

**schedule = []**

**def add\_surgery():**

**patient = input("Enter patient name: ")**

**time = input("Enter surgery time: ")**

**schedule.append({"Patient": patient, "Time": time})**

**print("Surgery scheduled successfully.\n")**

**def view\_schedule():**

**if schedule:**

**for s in schedule:**

**print(f"{s['Patient']} - Surgery at {s['Time']}")**

**else:**

**print("No surgeries scheduled.\n")**

**while True:**

**print("1. Add Surgery\n2. View Schedule\n3. Exit")**

**choice = input("Choose: ")**

**if choice == "1":**

**add\_surgery()**

**elif choice == "2":**

**view\_schedule()**

**elif choice == "3":**

**break**

**else:**

**print("Invalid choice.)**

**OUTPUT:**

**1. Add Surgery**

**2. View Schedule**

**3. Exit**

**Choose: 1**

**Enter patient name: Shalini**

**Enter surgery time: 10:30 AM**

**Surgery scheduled successfully.**

**Day 4 – ICU Monitoring in Action**

**Day 4 was about ICU wards and real-time patient monitoring. I simulated this in Python by creating a vitals monitor that tracks heart rates and temperatures, mimicking how critical data helps save lives.**

**import random**

**def monitor\_patient():**

**patient = input("Enter patient name: ")**

**heart\_rate = random.randint(60, 100)**

**temperature = random.uniform(36.5, 37.5)**

**print(f"{patient} - Heart Rate: {heart\_rate} bpm, Temperature: {temperature:.1f}°C")**

**while True:**

**print("1. Monitor Patient\n2. Exit")**

**choice = input("Choose: ")**

**if choice == "1":**

**monitor\_patient()**

**elif choice == "2":**

**break**

**else:**

**print("Invalid choice.\n")**

**OUTPUT:**

**Monitor Patient**

**2. Exit**

**Choose: 1**

**Enter patient name: Rahul**

**Rahul - Heart Rate: 85 bpm, Temperature: 36.8°C**

**Day 5 – Managing Diagnostic Records**

**Mammography, X-Rays, Ultrasound – I explored diagnostic tools and their role in quick patient diagnosis. Inspired, I created a Python program to log and view diagnostic reports digitally.**

**diagnostic\_records = {}**

**def add\_record(patient\_id, name, test\_type, result):**

**diagnostic\_records[patient\_id] = {**

**"Name": name,**

**"Test": test\_type,**

**"Result": result**

**}**

**print(f"Diagnostic record added for Patient ID: {patient\_id}")**

**def view\_records():**

**if not diagnostic\_records:**

**print("No diagnostic records found.")**

**return**

**print("\n--- Diagnostic Records ---")**

**for pid, details in diagnostic\_records.items():**

**print(f"Patient ID: {pid}")**

**for key, value in details.items():**

**print(f"{key}: {value}")**

**print("---------------------------")**

**add\_record(101, "John Doe", "X-Ray", "Fracture Detected")**

**add\_record(102, "Jane Smith", "Ultrasound", "Normal")**

**add\_record(103, "David Lee", "Mammography", "No Abnormalities")**

**view\_records()**

**OUTPUT:**

**Diagnostic record added for Patient ID: 101**

**Diagnostic record added for Patient ID: 102**

**Diagnostic record added for Patient ID: 103**

**--- Diagnostic Records ---**

**Patient ID: 101**

**Name: John Doe**

**Test: X-Ray**

**Result: Fracture Detected**

**---------------------------**

**Patient ID: 102**

**Name: Jane Smith**

**Test: Ultrasound**

**Result: Normal**

**---------------------------**

**Patient ID: 103**

**Name: David Lee**

**Test: Mammography**

**Result: No Abnormalities**

**---------------------------**

**Day 6 – Tracking Dosages with Code**

**Infusion Pumps and Syringe Pumps automate accurate dosages. I mirrored this by coding a dosage tracker in Python to help nurses record and review administered medicines.**

**dosages = []**

**def add\_dosage():**

**patient = input("Enter patient name: ")**

**medicine = input("Enter medicine name: ")**

**amount = input("Enter dosage amount (ml): ")**

**dosages.append({"Patient": patient, "Medicine": medicine, "Amount": amount})**

**print("Dosage recorded.\n")**

**def view\_dosages():**

**if dosages:**

**for d in dosages:**

**print(f"{d['Patient']} - {d['Medicine']} - {d['Amount']}ml")**

**else:**

**print("No dosages recorded.\n")**

**while True:**

**print("1. Add Dosage\n2. View Dosages\n3. Exit")**

**choice = input("Choose: ")**

**if choice == "1":**

**add\_dosage()**

**elif choice == "2":**

**view\_dosages()**

**elif choice == "3":**

**break**

**else:**

**print("Invalid choice.\n")**

**OUTPUT:**

**Add Dosage**

**2. View Dosages**

**3. Exit**

**Choose: 1**

**Enter patient name: Arjun**

**Enter medicine name: Paracetamol**

**Enter dosage amount (ml): 5**

**Dosage recorded.**

**Day 7 – Decoding Gas Cylinder Colors**

**Green for Oxygen, Black for Nitrogen. On Day 7, I learned about hospital gas cylinder color codes and created a Python solution that tells staff which gas each cylinder carries.**

**gas\_colors = {**

**"Green": "Oxygen",**

**"Black": "Nitrogen",**

**"Blue": "Nitrous Oxide",**

**"Grey": "Carbon Dioxide"**

**}**

**color = input("Enter cylinder color: ")**

**gas = gas\_colors.get(color, "Unknown Gas")**

**print(f"Cylinder Color: {color} - Gas Type: {gas}")**

**OUTPUT:**

**Enter cylinder color: Green**

**Cylinder Color: Green - Gas Type: Oxygen**

**Day 8 – Oxygen & Ventilator Simulation**

**In the ICU, I learned how ventilators and oxygen flow monitors work. To extend this learning, I designed a Python program to log and monitor patients’ oxygen saturation levels.**

**oxygen\_levels = {}**

**def update\_level():**

**patient = input("Enter patient name: ")**

**level = input("Enter oxygen level (%): ")**

**oxygen\_levels[patient] = level**

**print(f"Oxygen level updated for {patient}.\n")**

**def view\_levels():**

**if oxygen\_levels:**

**for p, l in oxygen\_levels.items():**

**print(f"{p}: {l}%")**

**else:**

**print("No oxygen levels recorded.\n")**

**while True:**

**print("1. Update Level\n2. View Levels\n3. Exit")**

**choice = input("Choose: ")**

**if choice == "1":**

**update\_level()**

**elif choice == "2":**

**view\_levels()**

**elif choice == "3":**

**break**

**else:**

**print("Invalid choice.\n")**

**OUTPUT:**

**1. Update Level**

**2. View Levels**

**3. Exit**

**Choose: 1**

**Enter patient name: Sneha**

**Enter oxygen level (%): 98**

**Oxygen level updated for Sneha.**

**Day 9 – Consolidating Patient Records**

**Day 9 focused on bringing together everything I learned. I built a Python system that consolidates patient details – from admissions to diagnostics into one streamlined view.**

**patients\_summary = []**

**def add\_patient():**

**name = input("Enter patient name: ")**

**age = input("Enter age: ")**

**issue = input("Enter health issue: ")**

**summary = input("Enter treatment summary: ")**

**patients\_summary.append({"Name": name, "Age": age, "Issue": issue, "Summary": summary})**

**print("Patient summary added.\n")**

**def view\_summaries():**

**if patients\_summary:**

**for p in patients\_summary:**

**print(f"{p['Name']} - {p['Age']} yrs - {p['Issue']} - Summary: {p['Summary']}")**

**else:**

**print("No patient summaries found.\n")**

**while True:**

**print("1. Add Patient\n2. View Summaries\n3. Exit")**

**choice = input("Choose: ")**

**if choice == "1":**

**add\_patient()**

**elif choice == "2":**

**view\_summaries()**

**elif choice == "3":**

**break**

**else:**

**print("Invalid choice.\n")**

**OUTPUT:**

**1. Add Patient**

**2. View Summaries**

**3. Exit**

**Choose: 1**

**Enter patient name: Karthik**

**Enter age: 45**

**Enter health issue: Diabetes**

**Enter treatment summary: Insulin therapy ongoing**

**Patient summary added.**

**Day 10 – Gratitude and Reflection**

**On my final day, I reflected on 10 days of intense learning and innovation. To summarize, I created a Python script listing all my key takeaways and the impact of healthcare IT on patient care.**

**learnings = [**

**"Biomedical devices understanding",**

**"Gas system handling",**

**"ICU monitoring simulations",**

**"Patient data management with Python"**

**]**

**print("10-Day Journey Summary:")**

**for i, item in enumerate(learnings, 1):**

**print(f"Day {i}: {item}")**

**OUTPUT:**

**10-Day Journey Summary:**

**Day 1: Biomedical devices understanding**

**Day 2: Gas system handling**

**Day 3: ICU monitoring simulations**

**Day 4: Patient data management with Python**