**Webcam Image Capture Backend Using Libraries And SocketIO**

## **Objective**

The goal of this project is to develop a Python backend system that:

* Connects to a normal webcam (USB camera or laptop integrated camera).
* Emits the frames in real-time to connected frontend clients using WebSocket communication (via Socket.IO).
* Saves each captured frame as a .jpg file locally for archival and analysis purposes.

**Image Processing : OpenCV**

1. **Client Connection :**

* The frontend client connects to the backend server using Socket.IO WebSocket protocol.
* When a client connects, the server logs the connection event.

1. **Starting Webcam Capture :** 
   * The backend starts a background thread that runs an asynchronous function (capture\_frames).
   * OpenCV accesses the default webcam using cv2.VideoCapture(0).

**Frame Capture and Processing**

* + OpenCV captures frames continuously in a loop.
  + Each frame is encoded as JPEG using cv2.imencode('.jpg', frame, [int(cv2.IMWRITE\_JPEG\_QUALITY), 50]) with 50% compression quality.
  + The JPEG binary is base64-encoded to create a string suitable for JSON transmission.

**Sending Frame to Client**

* + The base64 image data is sent to the connected client over the WebSocket using Socket.IO emit events.

**Saving Frame Locally**

* + Each frame is saved as a .jpg file with a unique timestamped filename in the captures/ directory.
  + Example filename: frame\_20250703\_102345\_123456.jpg

**Continuous Operation**

* + The system runs indefinitely, continuously capturing, sending, and saving frames until stopped.

****