**Code-**

**import cv2  
import winsound  
import tkinter as tk  
from tkinter import messagebox  
  
# Load the pre-trained Haar Cascade Classifiers for face and eyes  
face\_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade\_frontalface\_default.xml')  
eye\_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade\_eye.xml')  
  
  
def start\_detection():  
 # Try to open the camera  
 cap = cv2.VideoCapture(0) # Try camera index 0  
  
 if not cap.isOpened():  
 print("Error: Could not open video stream.")  
 messagebox.showerror("Error", "Could not access camera. Please check if it is in use or connected properly.")  
 return  
  
 drowsy\_count = 0 # Counter for drowsiness  
 THRESHOLD\_FRAMES = 20 # Frames before alarm triggers  
  
 while True:  
 ret, frame = cap.read()  
 if not ret:  
 break  
  
 gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)  
  
 # Detect faces in the grayscale image  
 faces = face\_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))  
  
 if len(faces) == 0:  
 drowsy\_count += 1 # Increment drowsy count if no faces detected  
 else:  
 # Loop through each face detected  
 for (x, y, w, h) in faces:  
 # Extract the region of interest (ROI) where the face is detected  
 roi\_gray = gray[y:y + h, x:x + w]  
  
 # Detect eyes within the face region  
 eyes = eye\_cascade.detectMultiScale(roi\_gray)  
  
 if len(eyes) == 0:  
 drowsy\_count += 1 # Increment drowsy count if no eyes detected  
 else:  
 drowsy\_count = 0 # Reset drowsy count if eyes are detected  
  
 # Draw rectangle around the face  
 cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)  
  
 # Draw rectangles around the eyes  
 for (ex, ey, ew, eh) in eyes:  
 cv2.rectangle(frame, (x + ex, y + ey), (x + ex + ew, y + ey + eh), (0, 255, 0), 2)  
  
 # If drowsy count exceeds the threshold, trigger the alarm  
 if drowsy\_count >= THRESHOLD\_FRAMES:  
 cv2.putText(frame, "WAKE UP!", (100, 100), cv2.FONT\_HERSHEY\_SIMPLEX, 1, (0, 0, 255), 2)  
 winsound.Beep(1000, 500)  
  
 cv2.imshow("Driver Alertness", frame)  
  
 if cv2.waitKey(1) & 0xFF == ord('q'):  
 break  
  
 cap.release()  
 cv2.destroyAllWindows()  
  
  
def show\_info():  
 messagebox.showinfo("Instructions", "Press 'q' to stop the detection.")  
  
  
# Create Tkinter GUI  
root = tk.Tk()  
root.title("Driver Alertness System")  
  
label = tk.Label(root, text="Driver Alertness Detection", font=("Arial", 14))  
label.pack(pady=10)  
  
start\_button = tk.Button(root, text="Start Detection", command=start\_detection, font=("Arial", 12))  
start\_button.pack(pady=5)  
  
info\_button = tk.Button(root, text="Instructions", command=show\_info, font=("Arial", 12))  
info\_button.pack(pady=5)  
  
exit\_button = tk.Button(root, text="Exit", command=root.quit, font=("Arial", 12))  
exit\_button.pack(pady=10)  
  
root.mainloop()**