Experiment-6.py

```python
# 18BCS6201-CV Practical-6 (Gauri Prabhakar) (AI-ML-2)(B)
# Aim: To implement pose detection using mediapipe in python and OpenCV.

# Importing necessary modules.
import cv2
import mediapipe as mp

# We will use 'drawing_utils' to draw the key points.
mp_drawing = mp.solutions.drawing_utils
mp_drawing_styles = mp.solutions.drawing_styles
# Importing 'mp.solutions.pose' to a variable 'mp_pose'.
mp_pose = mp.solutions.pose

# Creating a variable to store the video using the '.VideoCapture()' function.
cap = cv2.VideoCapture(r"C:\Users\gauri\Desktop\OpenCV Media\pose.mp4")

# Specifying the detection and tracking confidence uisng 'mp_pose.pose()'.
with mp_pose.Pose(
    min_detection_confidence=0.5, min_tracking_confidence=0.5) as pose:

    # While the video is running.
  while cap.isOpened():
    # Capturing the video frame by frame using the '.read()' method.
    success, image = cap.read()

    # To improve performance, marking the image as not writable and passing by reference.
    image.flags.writeable = False
    # Reading the frames and converting them to RGB.
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    # Detecting poses in the frame using the function 'pose.process()'.
    results = pose.process(image)

    # Drawing the pose annotations on the image.
    image.flags.writeable = True

    # Reading the frames and converting them to RGB.
    image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)

    # Connecting the key points using the function 'mp_drawing.draw_landmarks()'.
    mp_drawing.draw_landmarks(image, results.pose_landmarks, mp_pose.POSE_CONNECTIONS,
                              landmark_drawing_spec=mp_drawing_styles.get_default_pose_landmarks_style())

    # Rendering the video with effective Posetracking to the console by using the function '.imshow()'.
    cv2.imshow('Detecting poses using Mediapipe', image)

    # Setting up '.waitkey()' to wait for a specific time until any key is pressed and break the loop.
    # '.waitkey(1)' displays a frame for 1ms after which it moves to the next frame in the video.
    # Setting 'x' as the quitting button.
    if cv2.waitKey(5) & 0xFF == ord('x'):
      break

# Releasing the variable/object 'cap'.
cap.release()
```