Explanation of Face Verification System (OpenCV + LBPH)

Step-by-Step Breakdown of `verify_face()` Function

1. Load Attendee's Face Image

- Retrieves attendee info using Django's ORM.
- Fetches the path to the registered face image.

2. Convert Image to Grayscale and Resize

- Converts image to grayscale (LBPH works with grayscale).
- Resizes image to 200x200 for standard input.

3. Train LBPH Face Recognizer

- Initializes OpenCV's LBPH face recognizer.
- Trains it using the registered face with label '0'.

4. Open Webcam and Load Haar Cascade

- Loads Haar cascade for real-time face detection.
- Opens webcam by testing indexes 0 to 2.

5. Real-Time Verification Loop

- Captures frames from webcam.
- Converts each frame to grayscale.
- Detects faces using Haar cascade.
- For each face:
    - Resizes and compares using recognizer.predict().

6. Match Detection

- A match is found if confidence < 60.
- Confidence measures similarity (lower is better).

7. Visual Feedback (UI)

- Green box and "Match!" on success.
- Red box and "Not Match" if not.

8. Exit Conditions

- Breaks if face matched, 30s timeout, or user presses 'q'.

9. Cleanup

- Releases camera and closes OpenCV windows.

10. Exception Handling and Fallback

- Prints error trace if any exception occurs.
- Asks user to continue if verification fails.

Summary

Feature	Description
Face Detection	Haar Cascade
Face Recognition	LBPH (Local Binary Pattern Histogram)
Input	Grayscale 200x200 image
Output	Match decision based on confidence
Integration	Django ORM + Webcam + OpenCV