PROJECT OVERVIEW STATEMENT	Project Name: "Real-Time AI Face Verification System"	Student Name: KRUPALI SHINDE

Problem/Opportunity:

- Traditional face recognition systems are commonly used for user authentication in applications like attendance tracking, access control, and secure login.
- However, these systems are vulnerable to spoofing attacks, such as photos, videos, or deepfakes, which can easily bypass the system.
- Liveness detection is essential to ensure that the user is physically present and actively interacting with the system, not using a static image or recorded video.
- This project seeks to address these vulnerabilities by developing an AI-powered real-time face verification system with dynamic facial verification commands (e.g., blinking, smiling, or turning the head), ensuring higher security and preventing unauthorized access.

Goal:

The goal of this project is to design and implement a **Real-Time AI Face Verification System** that integrates **dynamic facial verification commands**. These commands (such as blinking, smiling, or head tilting) will be randomly generated and required to verify the user's identity, ensuring both **liveness detection** and **real-time interaction**. The system will ultimately enhance security by preventing spoofing and enabling robust authentication.

- **Specific**: Implement a face verification system with dynamic verification commands to detect liveness and prevent spoofing.
- **Measurable**: The system will successfully verify the user based on real-time action recognition (with a detection accuracy of 95% or higher).
- **Assignable**: The project will be carried out by Krupali Shinde.
- **Realistic**: The system will be implemented using accessible libraries such as OpenCV, TensorFlow, and Dlib, ensuring feasible development within the given resources.
- **Time-related**: The system will be completed within $2^{1/2}$ months.

Objectives:

Develop a Face Detection and Recognition System

- **Outcome**: To implement a real-time facial recognition system using OpenCV and Dlib.
- **Time Frame**: To be Completed within 2 weeks.
- **Measure**: Accuracy of 95% in face recognition.
- **Action**: Use face detection algorithms to track faces and recognize them in real time.

Implement Dynamic Command Generation for Verification

- **Outcome**: To generate random verification commands (e.g., blinking, turning the head).
- **Time Frame**: To be Completed within 3 weeks.
- **Measure**: The system successfully generates and processes some unique commands for verification.
- **Action**: Integrate AI models to analyze facial movements and recognize specific actions.

Integrate Liveness Detection

- **Outcome**: To prevent spoofing by ensuring the user performs real-time facial actions.
- **Time Frame**: To be Completed within 3 weeks.
- **Measure**: No successful spoofing attempts to be detected in 50 test cases.
- **Action**: Implement methods to detect blinks, head movement, and other spontaneous actions.

Develop User Interface (UI)

- **Outcome**: To create an intuitive and user-friendly interface for the face verification process.
- **Time Frame**: To be Completed within 2 weeks.
- **Measure**: UI is accessible and functional, with smooth user interaction.
- **Action**: Python's Flask for UI development.

Success Criteria:

Successful Authentication: The system must correctly verify users based on dynamic commands with an accuracy rate of 95% or higher.

Real-Time Performance: The system must perform the verification within 10 seconds for a seamless user experience.

Prevention of Spoofing: The system must successfully prevent spoofing attacks in 99% of test cases (e.g., using static images or videos).

User Experience: A smooth and responsive UI that allows users to perform the necessary actions without significant delay or frustration.

Assumptions, Risks, Obstacles:

1. **Assumptions**:
   - Users will be in a well-lit environment for accurate face detection.
   - The system will be used with a standard webcam or mobile camera.
   - Users are aware of the actions required (e.g., blinking, smiling) to perform the dynamic verification.

2. **Risks**:
   - **Low Accuracy in Poor Lighting**: The system may struggle to accurately detect faces or verify actions in low-light conditions.
   - **Model Performance**: The deep learning models used for real-time action detection might not be optimized for all users, leading to performance issues.
   - **System Latency**: Real-time processing may introduce latency, especially on less powerful devices.

3. **Obstacles**:
   - **Hardware Constraints**: If running on devices with lower processing power, the system might face delays or inaccuracies.
   - **User Variability**: Variations in facial features or expressions may affect the accuracy of action recognition (e.g., some users may find it difficult to perform specific actions like smiling).
   - **Environmental Factors**: Different lighting, angles, or background conditions could affect face detection accuracy.

Prepared By	Date	Approved By	Date
Krupali Shinde	06 February 2025		