# Face Recognition (PCA + ANN) – Complete Project Guide

## ■ Project Explanation Script (2–3 minutes)
“My project is an end-to-end Face Recognition system built using PCA for dimensionality reduction and ANN for classification.
Pipeline: Data Preprocessing → PCA (eigenfaces) → ANN training → Prediction.
Achieved ~50% accuracy. Modular design. Future scope: CNNs, bigger datasets, augmentation.”

## ■■ Project Workflow (Step-by-Step)
1. Preprocessing → Resize, Flatten, Database
2. PCA → Mean Face, Eigenfaces, Dim Reduction
3. Visualization → Save Mean Face, Eigenfaces
4. ANN → Dense layers, Softmax, Training
5. Save/Load Model → .h5 format
6. Testing → Test images, PCA, Predict

## ■ Example Terminal Output
■ Dataset loaded: 450 images (100x100)
■ Preprocessing Complete!
■ PCA Complete! Eigenfaces shape: (10000, 30)
■ Training ANN Classifier...
■ ANN Test Accuracy: ~50%
■ Model saved
■■ Test Image: face_3.jpg → Predicted: Disha

## ■ Interview Q&A;
Q1. What is PCA? Why used?
■ For dimensionality reduction & feature extraction.
Q2. What are Eigenfaces?
■ Principal components representing faces.
Q3. Why ANN not CNN?
■ Simpler demo, PCA+ANN works on small datasets.
Q4. Accuracy? Why low?
■ ~50%, limited dataset & ANN weakness.
Q5. Improvements?
■ CNNs, augmentation, more data.
Q6. Learnings?
■ Preprocessing, PCA, ANN, modular coding.

## ■ Final Project Structure
face_recognition_pca_ann/
■■■ data/ (faces/, test/)
■■■ results/ (mean_face.png, eigenfaces.png, model.h5)
■■■ src/ (preprocess.py, pca.py, visualize.py, ann.py, predict.py)
■■■ main.py
■■■ requirements.txt

# ■ 1-Page Summary: Face Recognition (PCA + ANN)

Goal: End-to-end face recognition using PCA (feature extraction) + ANN (classification).

Pipeline:
1■■ Preprocess → Resize (100x100), Flatten
2■■ PCA → Mean Face, Eigenfaces, Reduce 10k → 30–50
3■■ ANN → Train classifier on PCA features
4■■ Save Model (.h5)
5■■ Predict Test Images

Key Concepts:
- PCA = Dimensionality Reduction
- Eigenfaces = Principal Features of Faces
- ANN = Classifier with Dense + Softmax layers

Results:
- Accuracy ~50%
- Saved Mean Face & Eigenfaces
- Test predictions working

Interview Prep:
- Why PCA? → Reduce noise/dimension
- Why ANN? → Simple classifier on PCA features
- Why low accuracy? → Small dataset, ANN not ideal
- Improvements? → CNNs, Augmentation, More Data

Project Structure:
data/, results/, src/, main.py, requirements.txt

■ Covers end-to-end ML pipeline: Data → PCA → ANN → Predictions