Here's a detailed plan for the **Development Phase** of the Face Recognition Attendance System:

Development Phase Plan

Objective: Implement design specifications by developing and coding system components, including the face recognition module, web application, and database.

Sub-Phase 1: Setup and Preparation

Timeline: 1 Week
Activities:

1. Set up development environments:

    o Configure version control (e.g., Git/GitHub/GitLab).

    o Establish coding standards and documentation practices.

    o Set up local and cloud environments for testing and deployment.

2. Install necessary tools and libraries:

    o For face recognition: OpenCV, TensorFlow, or Dlib.

    o For web application: React/Angular (front-end), Django/Node.js (back-end).

    o Database: MySQL/PostgreSQL.

Deliverables:

- Development Environment Setup Report

- Installed Tools and Libraries Checklist

Sub-Phase 2: Face Recognition Module Development

Timeline: 3 Weeks
Activities:

1. Develop the face detection module:

    o Use pre-trained models (e.g., Haar cascades, Dlib, or YOLO).

    o Test with classroom camera feeds for robustness.

2. Implement face recognition:

    o Train the model on a dataset of student images.

    o Optimize for accuracy in diverse lighting and positioning scenarios.

3. Develop attendance matching logic:

    o Compare recognized faces with the student database.

o   Handle multi-face scenarios for large classrooms.

Deliverables:

- Face Detection Module

- Trained Face Recognition Model

- Attendance Matching Algorithm

Sub-Phase 3: Web Application Development

Timeline: 4 Weeks
Activities:

1. **Front-end Development**:

   o   Build intuitive user interfaces for the teacher dashboard and admin panel.

   o   Use React/Angular for dynamic and responsive design.

2. **Back-end Development**:

   o   Set up APIs for integrating face recognition, attendance data retrieval, and updates.

   o   Use Django/Node.js to handle back-end processes securely.

3. **Integration**:

   o   Connect the front-end with the back-end and ensure seamless data flow.

4. Implement authentication:

   o   Secure login for teachers and admins with role-based access.

Deliverables:

- Teacher Dashboard

- Admin Panel

- API Endpoints Documentation

- Authentication Module

Sub-Phase 4: Database Setup and Integration

Timeline: 2 Weeks
Activities:

1. Design and implement the database schema:

   o   Tables: Students, Attendance Records, Classes, Users (teachers/admins).

   o   Relationships and constraints for data integrity.

2. Implement database operations:

- o CRUD operations for student records, attendance logs, and class schedules.

- o Optimize queries for speed and scalability.

3. Integrate with other components:

- o Ensure seamless data flow between the face recognition module and the database.

Deliverables:

- Fully Functional Database

- Data Integration Logs

- Database Schema Documentation

Sub-Phase 5: Security Implementation

Timeline: 1 Week
Activities:

1. Implement security protocols:

- o Encrypt sensitive data (e.g., student records).

- o Use HTTPS for secure data transmission.

- o Implement OAuth2.0 or similar for secure user authentication.

2. Conduct vulnerability assessments:

- o Test for SQL injection, cross-site scripting (XSS), and other security risks.

Deliverables:

- Encrypted Database

- Secured Web Application

- Vulnerability Assessment Report

Sub-Phase 6: Unit Testing

Timeline: 1 Week
Activities:

1. Conduct unit tests for each component:

- o Face recognition module: Validate detection and recognition accuracy.

- o Web application: Test front-end and back-end functionalities.

- o Database: Test CRUD operations and query performance.

2. Document test cases, results, and fixes.

Deliverables:

- Unit Test Reports

- Updated Codebase with Bug Fixes

Total Timeline for Development Phase: 8-10 Weeks

Roles and Responsibilities

1. **Developers**: Work on front-end, back-end, and face recognition components.

2. **Data Scientists**: Train and fine-tune face recognition models.

3. **QA Testers**: Conduct unit testing and ensure functionality for each module.

4. **Project Manager**: Oversee timelines, deliverables, and team coordination.

This detailed development plan ensures that each component is implemented systematically and aligns with overall project goals. Let me know if you need further refinements!