Web application development plan

The provided document contains a comprehensive development plan for a face recognition attendance system. Here's how to begin the web application development portion of the project, following the specified roadmap:

Preparation Phase (Week 1)

1. **Set Up Development Environments:**

   o Configure version control using GitHub/GitLab.

   o Define coding standards and documentation guidelines.

   o Prepare local (e.g., VS Code, Node.js, Python) and cloud environments (AWS/GCP/Azure).

2. **Install Necessary Tools and Libraries:**

   o Front-end: React or Angular.

   o Back-end: Django or Node.js.

   o Database: MySQL/PostgreSQL.

   o Testing tools: Postman for APIs, Jest or Mocha for testing.

Sub-Phase 3: Web Application Development (Weeks 5-8)

1. Front-End Development:

- **Build Interfaces**:

   o Design the **teacher dashboard** with real-time attendance metrics.

   o Create an **admin panel** for user management and system settings.

- **Responsive Design**:

   o Use Material-UI/Bootstrap for styling.

   o Ensure compatibility across browsers and devices.

2. Back-End Development:

- **Set Up APIs**:

   o Develop endpoints for face recognition integration, attendance data retrieval, and CRUD operations.

   o Example technologies: RESTful API using Django REST Framework or Express.js.

- **Secure Back-End Logic**:

   o Handle authentication and authorization with JWT or OAuth2.0.

3. Integration:

- Connect the front-end to the back-end using REST/GraphQL APIs.

- Validate seamless data flow and error handling between components.

4. Authentication:

- Implement secure login features with role-based access.

- Use libraries like Django-Allauth or Passport.js for security layers.

Tips for Efficient Development

1. **Iterative Development**:

    o Build and test each module (e.g., login, dashboard) incrementally.

    o Ensure frequent code reviews and integration.

2. **Collaborative Tools**:

    o Use project management tools like Jira or Trello.

    o Maintain communication via Slack or MS Teams.

3. **Unit Testing**:

    o Employ frameworks like Jest or PyTest to test individual modules.

    o Test against mock data for reliability.