Implementation Plan

Phase 1: Core Features

1. **User Authentication and Authorization**
    - Implement user registration, login, and role-based access control.
    - Create database schemas for user and role management.
2. **Employee Information Management**
    - Develop forms and interfaces for adding and editing employee profiles.
    - Implement functionality for document uploads.
3. **Basic UI/UX Design**
    - Design the basic layout and navigation of the application.

Phase 2: Interactive Features (out of scope)

1. **Attendance Tracking**
    - Implement check-in/check-out functionality.
    - Develop leave management interfaces and approval workflows.
2. **Performance Evaluation**
    - Create interfaces for setting goals and tracking performance.
    - Implement functionality for recording performance reviews.
3. **Payroll Management**
    - Develop interfaces for managing salary details.
    - Implement payslip generation and download functionality.

Phase 3: Advanced Features (out of scope)

1. **Reporting and Analytics**
    - Implement reporting features for attendance, performance, and leave statistics.
    - Develop graphical representations of data for better insights.
2. **Notifications and Announcements**
    - Implement email and in-app notifications for important updates.
3. **Performance Optimization and Security Enhancements**
    - Optimize the application for performance and scalability.
    - Implement advanced security measures to protect user data.

Phase 4: Testing and Deployment

- **Unit Testing:** Test individual functionalities of the application to ensure they work as expected.
- **Integration Testing:** Test how different parts of the application work together seamlessly.

- **User Acceptance Testing (UAT):** Get feedback from potential users to identify usability issues and ensure the application meets their needs.
- **Deployment:** Deploy the application to a web server for user access. Consider cloud platforms like AWS or Google Cloud Platform for scalability and reliability.

Development Workflow

1. **Requirement Analysis**: Gather detailed requirements and create a project plan.
2. **System Design**: Design the system architecture, database schema, and API endpoints.
3. **Implementation**: Develop the application in phases, starting with the core features.
4. **Testing**: Conduct unit testing, integration testing, and user acceptance testing.
5. **Deployment**: Deploy the application to a cloud platform (AWS, Azure, GCP) and configure the necessary services.
6. **Maintenance and Updates**: Regularly update the platform based on user feedback and add new features.

Data Model

We are using AWS Dynamodb as a persistent data storage. Dynamodb is a nosql database. In a NoSQL database like DynamoDB, we'll able to organize our data in tables, with items (rows) and attributes (columns). DynamoDB is schema-less for the most part, but we need to define the primary key and optionally secondary indexes for efficient querying.

Table Schema

1. Employees Table

- **Primary Key**: `EmployeeID` (Partition Key), `Email` (Sort Key)
- **Attributes**:
  - `EmployeeID` (String): Unique identifier for each employee.
  - `Name` (String): Full name of the employee.
  - `Email` (String): Email address of the employee.
  - `Position` (String): Job position of the employee.
  - `Department` (String): Department the employee belongs to.

- `DateOfJoining` (String): Date the employee joined the company.

User management (AWS Cognito)

In AWS Cognito, you can define these roles using Cognito User Pools and Identity Pools. Here's how you can set it up:

1. **Create User Pool**:
   - Define attributes for user registration (e.g., email, username, role).
   - Configure sign-up and sign-in settings.
2. **Create Identity Pool**:
   - Link the user pool to the identity pool.
   - Define roles and policies for authenticated users.
3. **Assign Roles**:
   - Use AWS IAM to create roles with specific permissions.
   - Map these roles to users in Cognito based on their attributes (e.g., role).
   - Roles (groups): admin, HR, manager, employee.

Architecture Diagram