

# Population Management System Website Project – Cambodia: Empowering Commune-Level Data Administration

Enhancing local data management and  
administrative efficiency

# Meeting Program

- Project Overview
- Key Modules and Features
- Technology Stack
- System Architecture
- Database Schema Overview
- Folder Structures (Frontend and Backend)
- Multi-Tenancy and Access Control
- Reporting and Analytics
- Deployment Strategy

# Project Overview



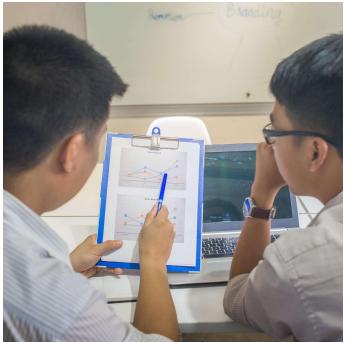
## Streamlining Data Management

The project focuses on modernizing population data management for enhanced efficiency and accuracy.



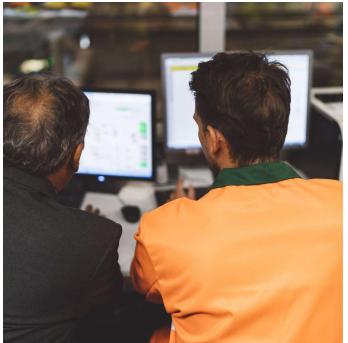
## Facilitating Data Entry

Facilitating easy and accurate data entry to keep population records up to date.



## Empowering Local Authorities

Empowering local authorities to use accurate data for effective decision-making at commune levels.



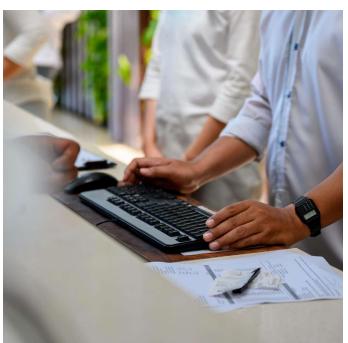
## Primary User Group

Commune-level officials are the main users responsible for managing local population data and administration.



## System Usability

The system is designed to fit their specific workflows, improving ease of use and efficiency in daily tasks.



## Data Accuracy and Governance

Enhanced data accuracy supports better decision making and governance at the local level.



# Scope and Anticipated Impact

## Comprehensive Data Coverage

The system manages population records and identity documents integrating multiple national databases for unified access.

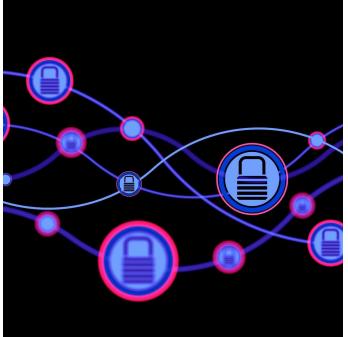
## Enhanced Data Reliability

Improving accuracy and consistency of data to support administrative decision-making and public services.

## Support for National Development

The system supports Cambodia's development goals by streamlining administration and enabling better planning.

# Key Modules and Features



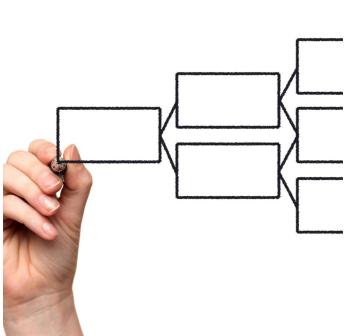
## Tracking Population Laws

The module tracks relevant population laws and regulations to ensure all legal requirements are monitored consistently.



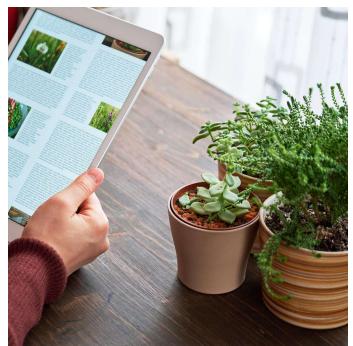
## Ensuring Compliance

It facilitates compliance with population legislation by managing updates and aligning with current legal standards.



## Legislation Updates

The system supports seamless updates to legislation affecting population data management to keep information current.



## Family Unit Data Management

Modules facilitate detailed management of family unit information for accurate household records.

## Residential Card Issuance

Modules handle issuance and tracking of residential cards ensuring organized identification at local levels.



## Commune Level Organization

These systems support accurate and systematic household record-keeping at the commune administrative level.



# National SIM Identity Card and Death Registration

## **SIM Identity Card Management**

Facilitates the issuance and management of national SIM identity cards ensuring secure and verified user identification.

## **Accurate Death Registration**

Ensures accurate and timely registration of deaths to maintain reliable and updated population data.



# Integration with NCDD Cambodia Communal Database

## Seamless Data Exchange

Integration enables smooth and uninterrupted exchange of data with the communal database for improved communication.

## Data Synchronization

Synchronization ensures that data is consistently updated across systems, reducing errors and discrepancies.

## Enhanced Data Reliability

Reliable integration supports accurate and trustworthy data management across platforms.



# Technology Stack

# Frontend: Next.js, React TSX, Shadcn UI, Tailwind CSS



## Next.js Framework

Next.js enables server-side rendering and static site generation for optimized React applications.

## React with TSX

React combined with TSX ensures component-based UI with type safety for better code quality.

## Shadcn UI Components

Shadcn UI provides customizable and reusable UI components to speed up frontend design.

## Tailwind CSS Styling

Tailwind CSS offers utility-first styling for rapid and responsive frontend development.

# Backend: Node.js TSX, PHP 4.x Native, Python 3.12.7



## Node.js with TSX

Node.js with TSX powers modern server-side logic and enables efficient handling of asynchronous operations.

## Legacy PHP 4.x Components

Legacy PHP 4.x components maintain existing integrations and support older backend functionalities.

## Python 3.12.7 for Processing

Python 3.12.7 handles specialized processing tasks and supports advanced backend operations.

# Databases: MySQL, PostgreSQL, MongoDB

## **MySQL and PostgreSQL**

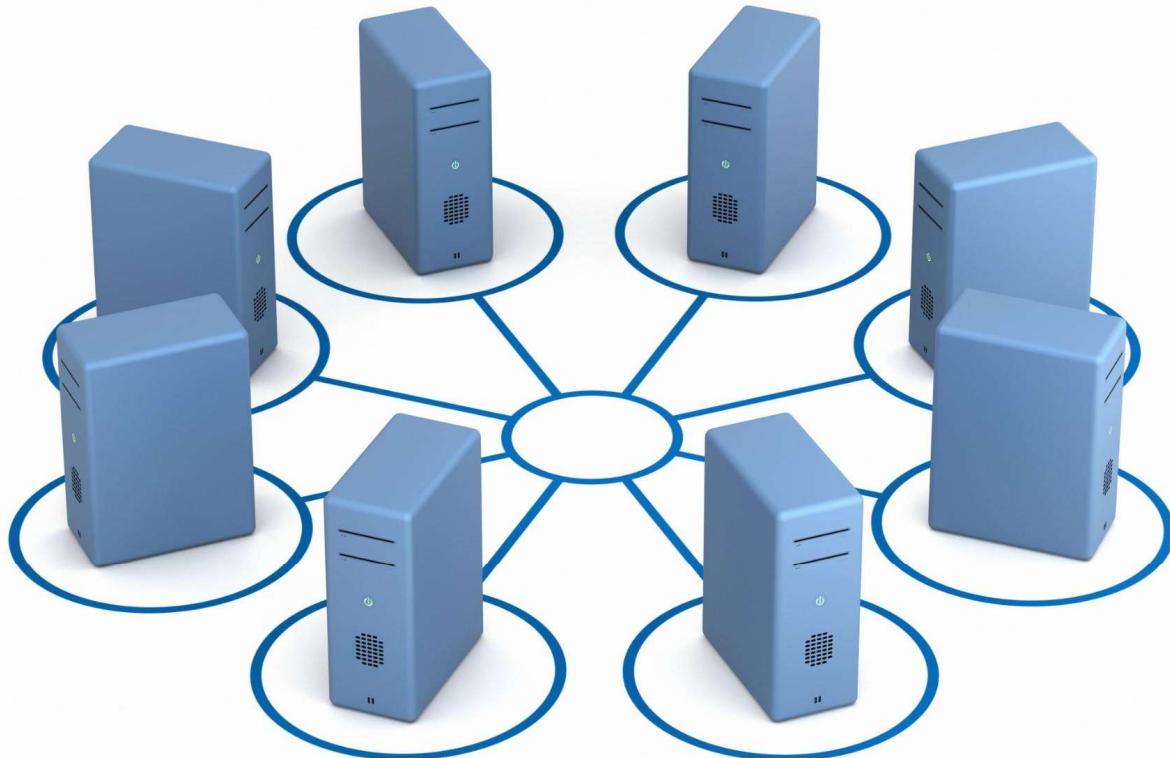
MySQL and PostgreSQL are relational databases used for structured data storage with strong consistency and ACID compliance.

## **MongoDB for Document Storage**

MongoDB provides flexible, scalable document-oriented storage suited for unstructured or semi-structured data.



# System Architecture



# High-Level Architectural Diagram

## System Components Overview

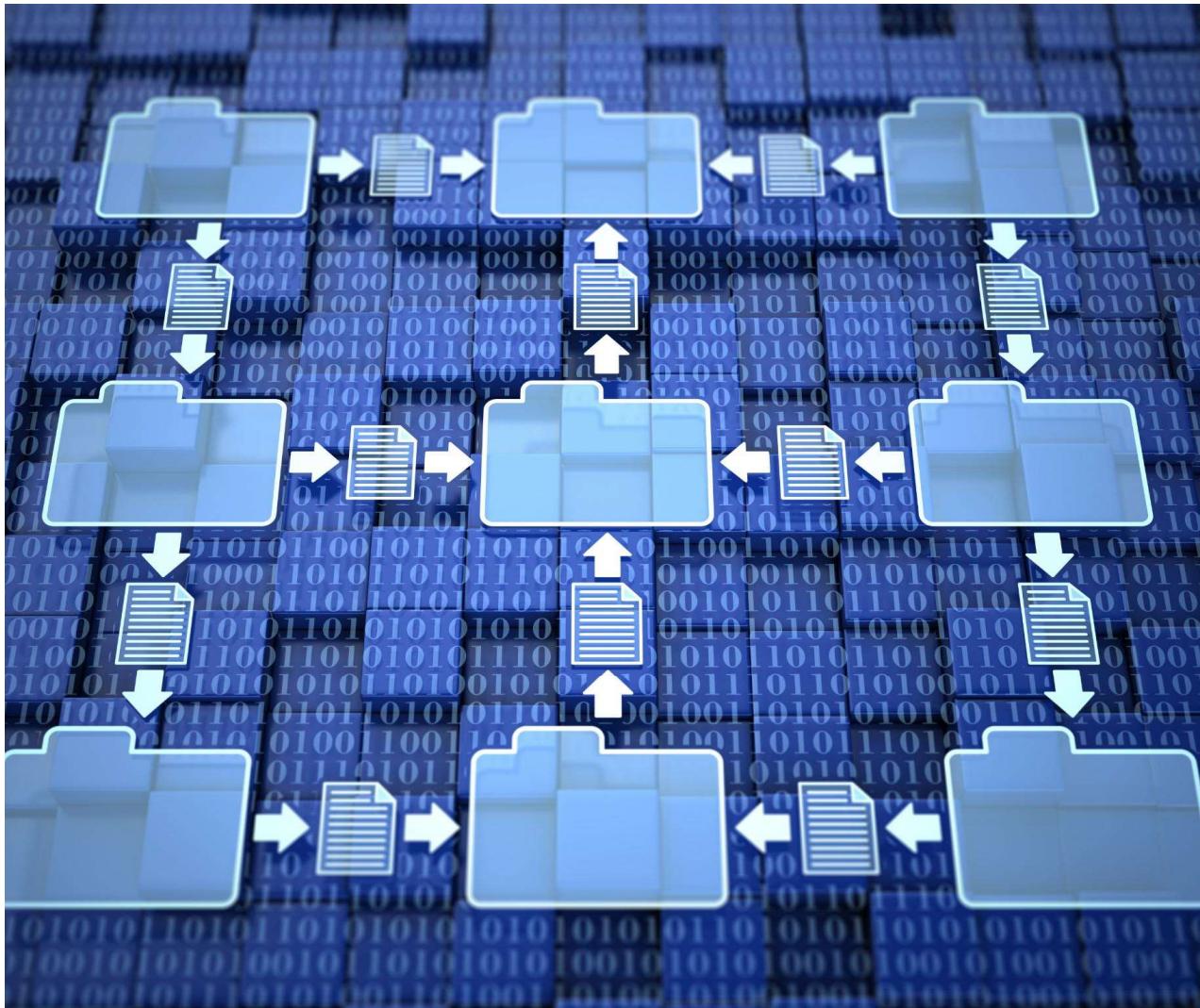
The diagram visually represents major system components and their interactions.

## Data Flow Illustration

Shows how data moves between frontend, backend, databases, and external systems.

## Control Flow Representation

Illustrates control signals and coordination among system parts for processing tasks.



# Module Interactions and Data Flows

## Module Communication

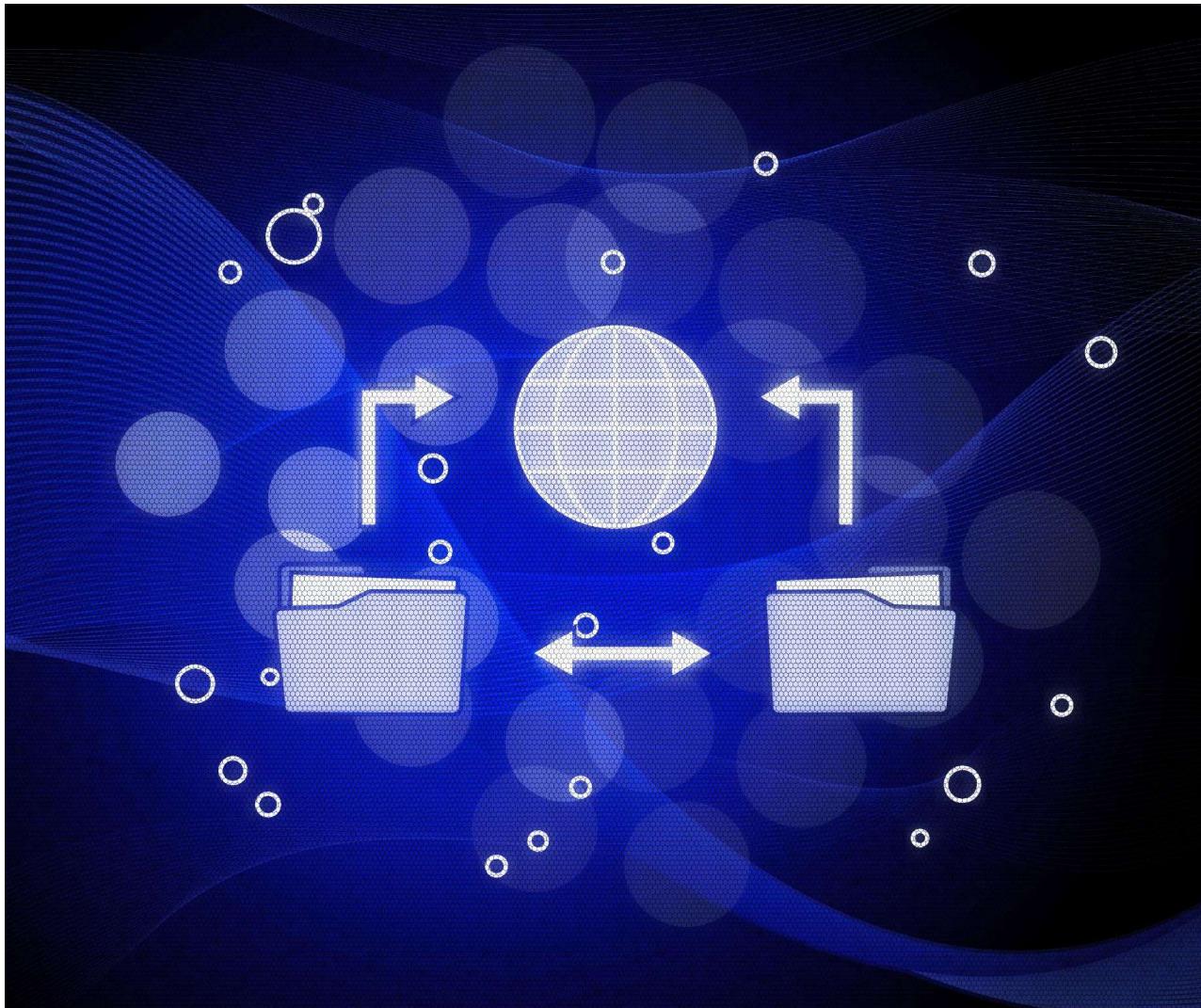
Modules communicate internally through defined interfaces to ensure coherent operations and data sharing.

## Data Processing Sequence

Data processing follows a sequence of steps that preserves data integrity and ensures timely responsiveness.

## Maintaining System Integrity

System integrity is maintained by validating data at each processing stage and handling errors effectively.



# Integration Points with External Systems

## Interfaces and APIs

Interfaces and APIs enable seamless communication between internal and external third-party systems for data exchange.

## Third-Party Integration

Integration with external systems like communal databases ensures up-to-date and synchronized population data.

# Database Schema Overview

# Entity Relationship Diagrams

## Representation of Entities

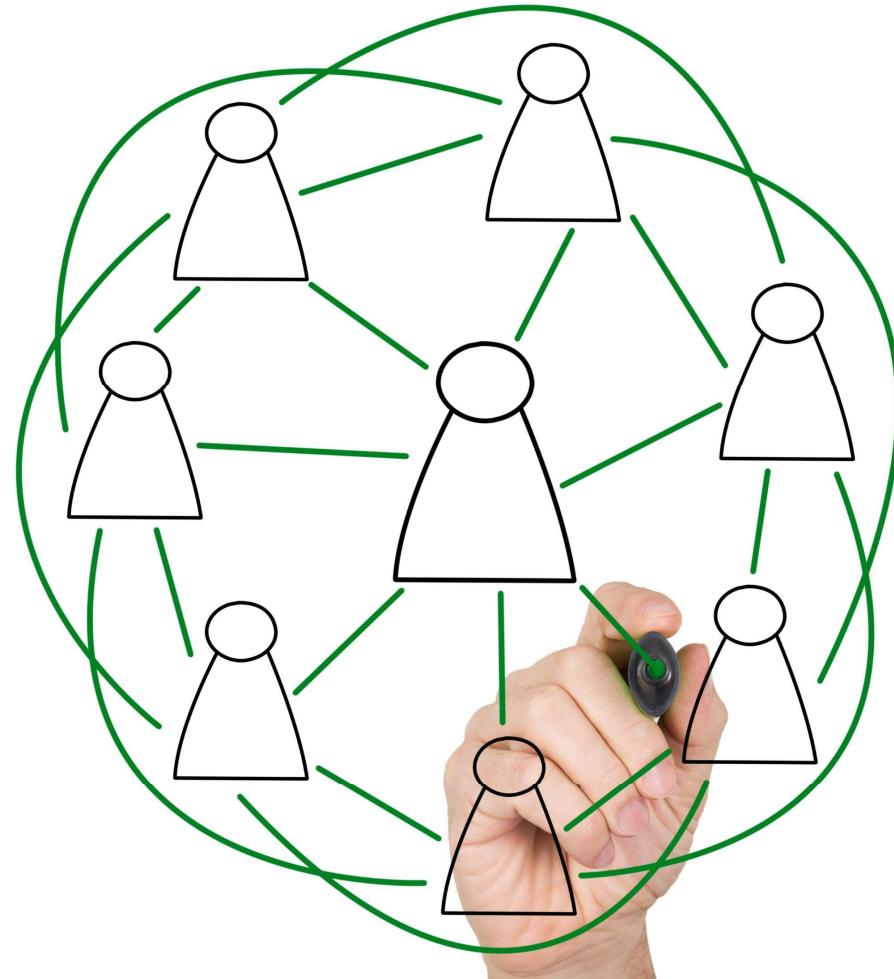
ER diagrams depict entities such as individuals, families, residences, and legal documents clearly within data models.

## Relationships Between Entities

The diagrams show how different entities relate to each other, aiding in understanding data structure and connections.

## Clarity in Data Structure

ER diagrams provide clarity by visually mapping complex relationships within data systems for better comprehension.





# Key Tables for Population Data

## Population Records Storage

Critical tables store detailed population records vital for demographic analysis and planning.

## Identity Information Management

Tables contain identity information essential for accurate citizen identification and verification.

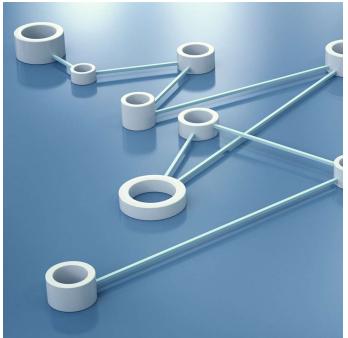
## Registration Data Handling

Registration data tables support commune-level management through organized data collection and storage.



## Reducing Data Redundancy

Normalization minimizes duplicate data, optimizing storage and improving database efficiency.



## Ensuring Data Consistency

Normalization maintains data accuracy and uniformity across the database tables.



## Securing Data Reliability

Normalization processes support reliable data retrieval and trustworthy information management.

# Folder Structures (Frontend and Backend)

# Frontend Folder Organization (Next.js & React)

## **Components Folder**

Holds reusable UI components to promote modularity and maintainability in frontend codebases.

## **Pages Folder**

Contains page-level components that define routes and layout structure within the application.

## **Styles Folder**

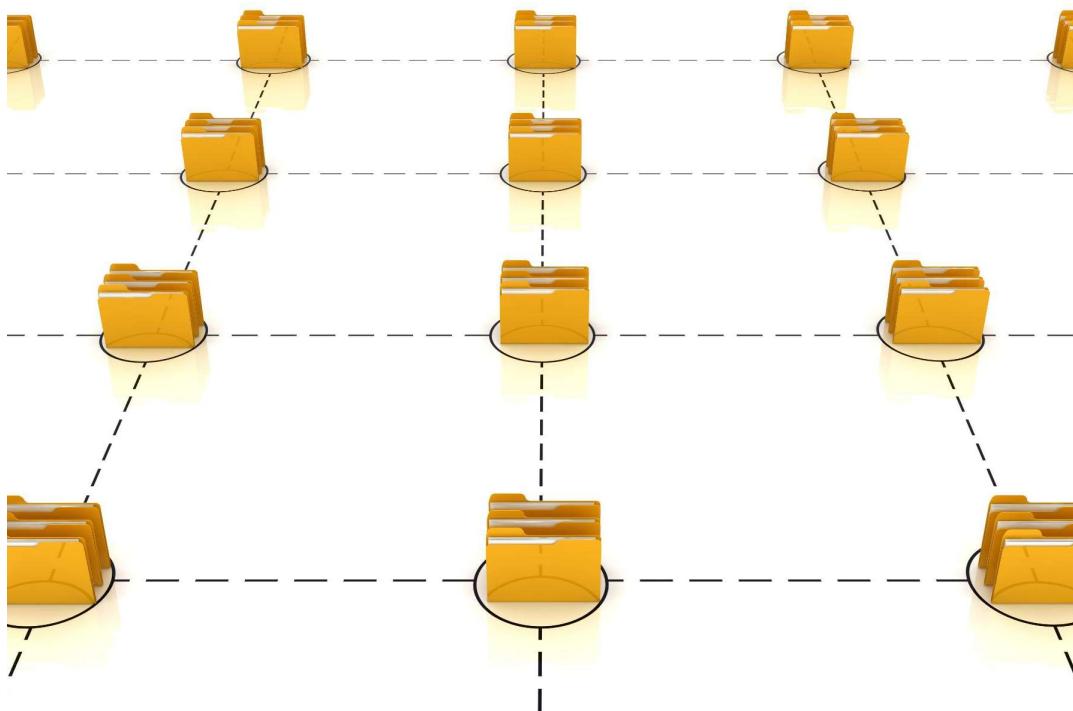
Stores CSS or styling files to maintain consistent design across the application.

## **Utilities Folder**

Includes helper functions and utilities that support application logic and code reuse.



# Backend Folder Organization (Node.js, PHP, Python)



## Service Separation

Services are organized into distinct folders by language to improve maintainability and clarity in backend operations.

## Script Management

Scripts are separated by programming language, helping developers easily locate and update code components.

## Legacy Code Handling

Legacy code is stored in dedicated directories to prevent interference with current services and streamline updates.

# Best Practices for Maintainability

## Coding Conventions

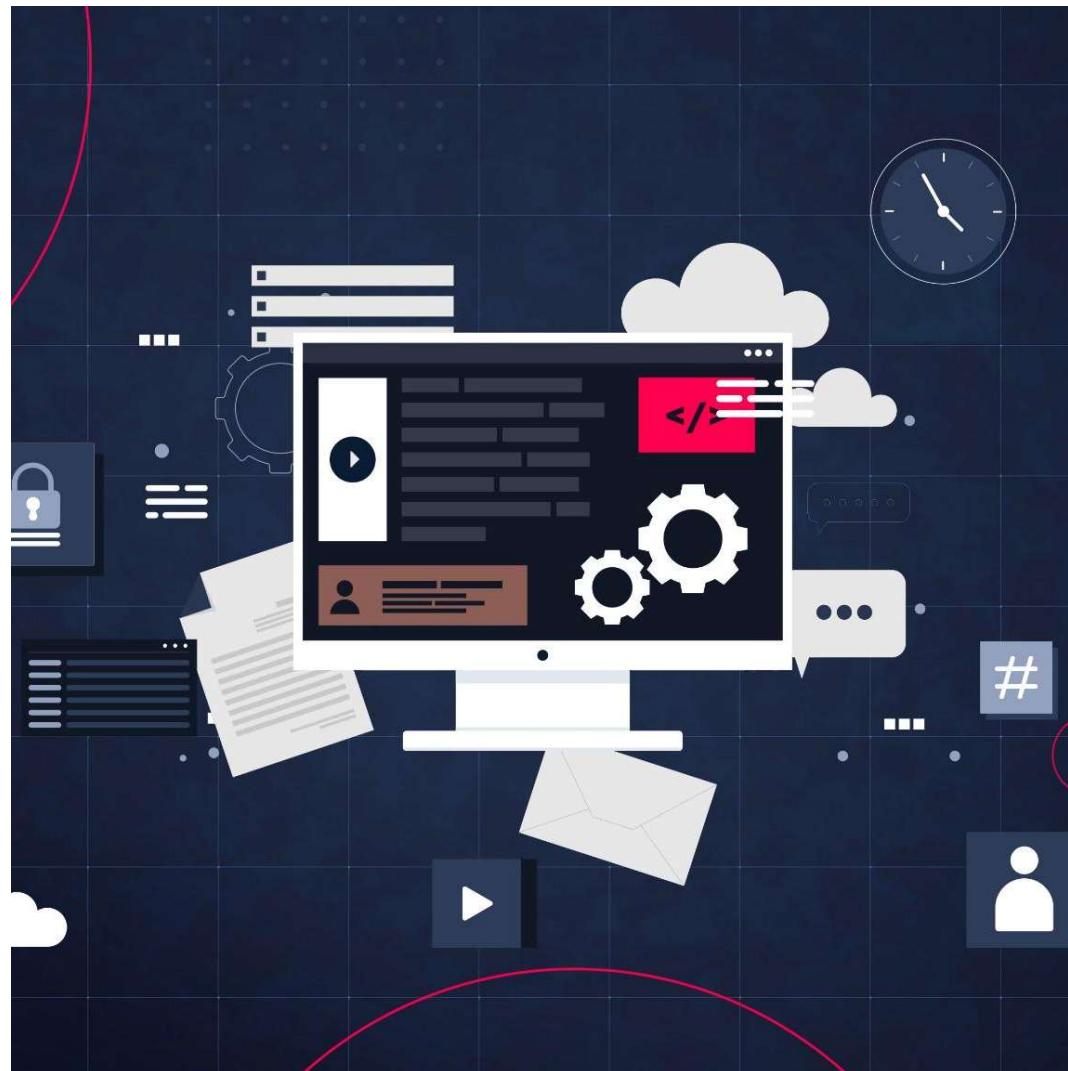
Consistent coding conventions improve readability and simplify maintenance across development teams.

## Documentation Standards

Clear documentation ensures understanding and ease of updates by current and future developers.

## Modular Design Principles

Modular design breaks systems into manageable components, facilitating updates and scalability.



# Multi-Tenancy and Access Control

# Multi-Tenancy Model for Communes



## **Data Isolation**

The model ensures that each commune's data is securely separated from others, preventing cross-access.

## **Configuration Independence**

Each commune can independently manage its settings and configurations without affecting others.

## **Secure Shared System**

The architecture supports multiple communes operating securely on a single shared infrastructure.



# Role-Based Access Control Mechanisms

## Role Assignment

Users are assigned specific roles that define their permissions within a system, ensuring organized access management.

## Permission Enforcement

Role-based access control enforces security policies by restricting data access and modifications to authorized roles only.



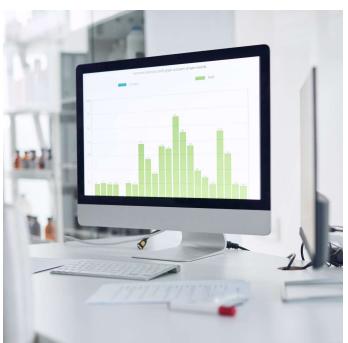
## Data Encryption

Data encryption secures sensitive population data by converting it into unreadable formats to unauthorized users.



## Authentication Protocols

Authentication protocols verify user identities to prevent unauthorized access to protected information.



## Audit Logging

Audit logging tracks system activities to monitor security events and ensure data integrity.

# Reporting and Analytics

# Population Statistics Dashboards

## Real-time Population Data

Dashboards offer up-to-date statistics on population size and growth to support timely decision-making.

## Demographic Trends Analysis

Visualization of age, gender, and other demographics helps identify social and economic trends.

## Planning Support for Authorities

Accurate data aids commune authorities in resource allocation and community planning.





## Custom Report Generation

### Tailored Report Creation

Users create reports customized to specific data and criteria matching administrative needs effectively.

### Efficiency in Reporting

Custom report generation streamlines administrative and government reporting processes, saving time.



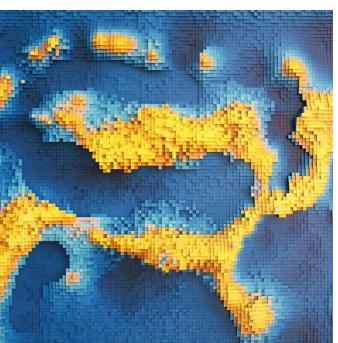
## Charts for Data Presentation

Charts help to visually display data trends and comparisons effectively for better understanding.



## Graphs for Analysis

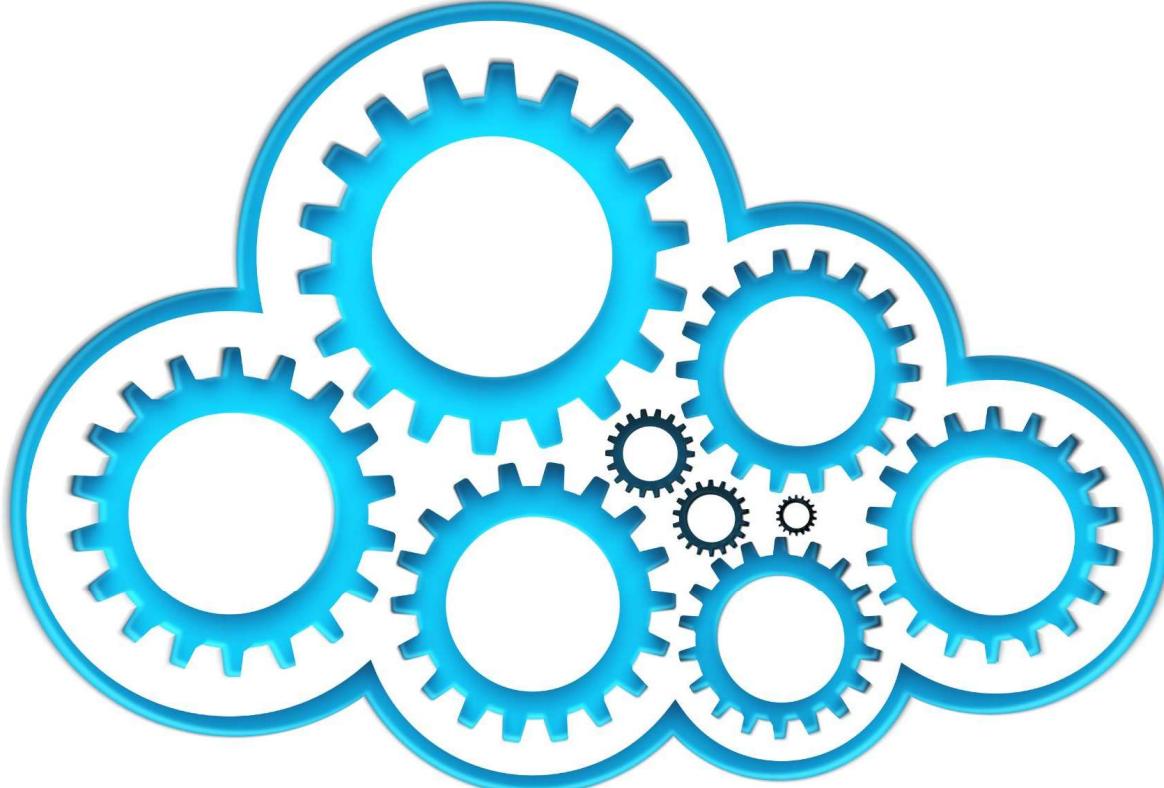
Graphs are used to illustrate relationships and patterns in complex datasets clearly.



## Maps for Spatial Data

Maps visualize geographical distribution of data, aiding in spatial analysis and interpretation.

# Deployment Strategy



# Deployment Pipelines and Environments

## Automated Build Process

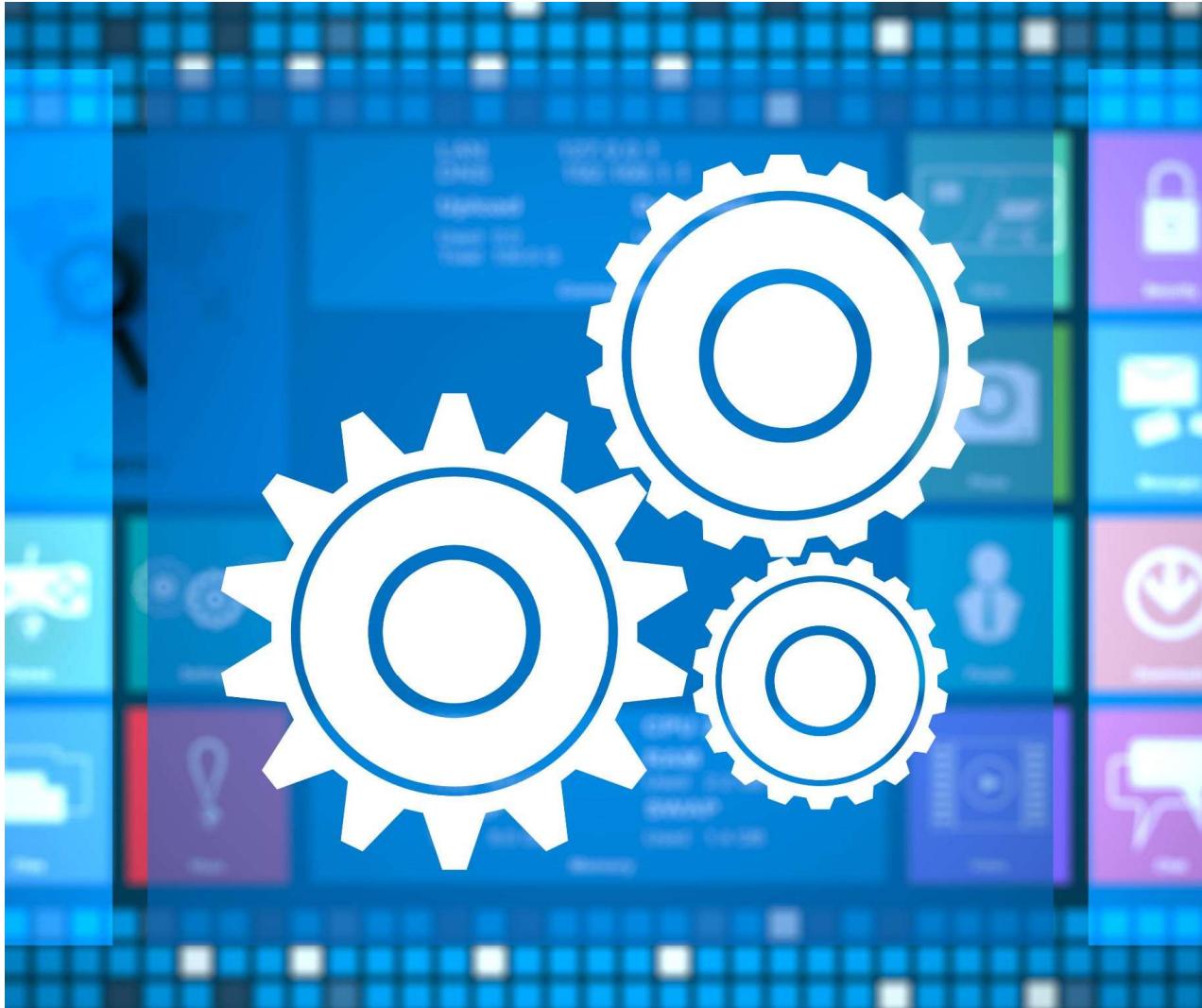
Automated pipelines compile and build code efficiently ensuring consistency across all environments.

## Testing Environments

Testing environments validate code quality and functionality before deployment to production.

## Deployment to Production

Code is deployed to production environments reliably using automated deployment pipelines.



# Continuous Integration / Continuous Deployment (CI/CD)

## Frequent Reliable Updates

CI/CD workflows enable frequent and reliable software updates to meet user needs quickly and efficiently.

## Minimal Downtime

Deployment strategies in CI/CD minimize downtime, ensuring continuous application availability and smooth user experience.

## Rapid Feedback Loops

CI/CD integrates testing and monitoring for rapid feedback, allowing quick detection and resolution of issues.

# Scalability and Reliability Measures

## Load Balancing

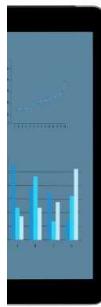
Load balancing distributes network traffic evenly to optimize resource use and prevent overloads.

## Redundancy

System redundancy ensures backup components are available to maintain service during failures.

## Monitoring

Continuous monitoring tracks system health and alerts to potential performance issues promptly.



# Conclusion

## **Empowering Commune Authorities**

The system equips local authorities in Cambodia with tools to manage population data effectively and efficiently.

## **Secure Data Management**

The platform ensures secure handling and protection of sensitive population data through advanced technology.

## **Supporting National Goals**

Improved population data administration supports Cambodia's broader national development and planning objectives.