INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Database Management System Laboratory

Mini-Project Report

Gram Panchayat Management System

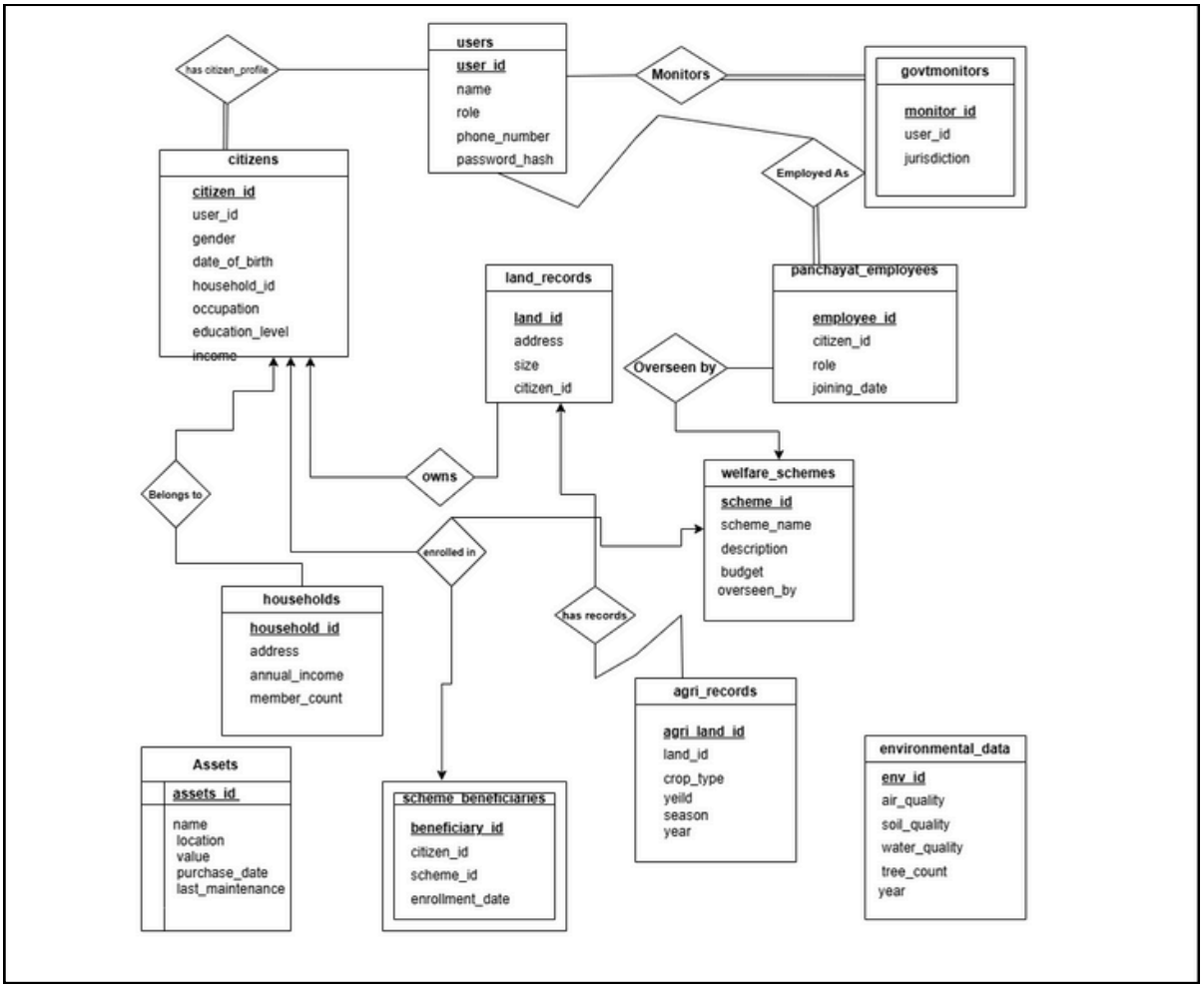
MyPanchayat

Advisor: Prof. Pabitra Mitra, Prof. KS Rao

Member List

Name	Roll Number
Harsh Chattar	22CS30028
Ansh Sahu	22CS30010
Sayali	22CS10034
Anand Malav	22CS10008
Abhinav Kumar Singh	22CS30005

Entity-Relational Diagram



The image is an Entity-Relationship (ER) diagram representing a database schema for managing citizen records, land ownership, welfare schemes, and government monitoring.

Key Entities and Relationships:

1. Users & Citizens: Users have a profile linked to citizens with details like gender, date of birth, and occupation.
2. Households & Assets: Citizens belong to households, which track income and members. Assets like property and valuables are also recorded.
3. Land Records & Agriculture: Citizens own land, with agricultural records keeping track of crops and yields.
4. Welfare Schemes: Citizens enroll in government welfare schemes, overseen by government monitors.
5. Government Monitoring: Panchayat employees and monitors oversee land and welfare schemes.
6. Environmental Data: Tracks air, soil, and water quality.

This schema supports governance, land management, and citizen welfare tracking.

Table Schema

agri_records
• agri_land_id: Unique ID (Primary Key)
• land_id: Unique ID (Refer to: land_records.land_id)
• crop_type: Text (e.g., “corn”, “wheat”, “soybeans”, “potatoes”, “tomatoes”, “carrots”, “cabbage”, “peas”, “lettuce”, “rice”)
• yield: Integer (Must be $\geqslant 0$)
• season: Text (e.g., “Monsoon”, “Summer”, “Winter”)
• year: Integer

assets
• asset_id: Unique ID (Primary Key)
• name: Text (Max 255 characters)
• type: Text (Max 100 characters)
• location: Text (Max 255 characters) (Optional)
• value: Numeric (15,2) (Optional)
• purchase_date: Date (Optional)
• last_maintenance: Date (Optional)

census_data
• census_id: Unique ID (Primary Key)
• year: Integer
• births: Integer
• deaths: Integer
• marriages: Integer
• divorces: Integer
• male_population: Integer
• female_population: Integer
• total_population: Integer (Auto-generated as male_population + female_population)

Table Schema

citizens
• citizen_id: Unique ID (Primary Key)
• user_id: Unique ID (Refer to: users.user_id) (Unique)
• gender: Text (e.g., “Male”, “Female”, “Others”)
• date_of_birth: Date (Must be ⩽ Current Date)
• household_id: Unique ID (Refer to: households.household_id)
• occupation: Text (Max 50 characters) (Optional)
• education_level: Text (e.g., “High School Diploma”, “Bachelor’s Degree”, “Master’s Degree”, “PhD”, “Technical Certification”, “None”)
• income: Integer (Must be ⩾ 0) (Optional)

environmental_data
• env_id: Unique ID (Primary Key)
• air_quality: Numeric (5,2) (Range: 0 - 500)
• soil_quality: Numeric (5,2) (Range: 0 - 100)
• water_quality: Numeric (5,2) (Range: 0 - 100)
• tree_count: Integer (Must be ⩾ 0)
• year: Integer (Must be ⩽ Current Year)

govtmonitors
• monitor_id: Unique ID (Primary Key)
• user_id: Unique ID (Refer to: users.user_id) (Unique)
• jurisdiction: Text (e.g., “District”, “Block”, “Village”, “Ward”, “Taluka”, “Tehsil”, “Subdivision”, “Zone”, “Cluster”)

households
• household_id: Unique ID (Primary Key)
• address: Text
• annual_income: Integer (⩾ 0)
• member_count: Integer (> 0)

Table Schema

land_records
• land_id: Unique ID (Primary Key)
• address: Text
• size: Numeric (4,2) (> 0)
• citizen_id: Integer (Foreign Key → citizens.citizen_id, CASCADE DELETE)

panchayat_employees
• employee_id: Unique ID (Primary Key)
• citizen_id: Integer (Foreign Key → citizens.citizen_id, CASCADE DELETE, UNIQUE)
• role: Text (Sarpanch, Panchayat Secretary, Village Development Officer, etc.)
• joining_date: Date (⩽ CURRENT_DATE)
• overseen_by: Foreign Key in welfare_schemes (SET NULL on DELETE)

scheme_beneficiaries
• beneficiary_id: Unique ID (Primary Key)
• citizen_id: Integer (Foreign Key → citizens.citizen_id, CASCADE DELETE)
• scheme_id: Integer (Foreign Key → welfare_schemes.scheme_id, CASCADE DELETE)
• enrollment_date: Date (Default: CURRENT_DATE, Not NULL)

tables
• name: VARCHAR(50) (Primary Key, Not NULL)
• title: VARCHAR(100) (Not NULL)
• description: TEXT (Optional)

Table Schema

users
- user_id: INTEGER (Primary Key, Auto-increment)
- name: VARCHAR(100) (Not NULL)
- role: VARCHAR(20) (Not NULL, with constraints)
- phone_number: VARCHAR(15) (Unique, Not NULL)
- password_hash: TEXT (Not NULL)

welfare_schemes
- scheme_id: INTEGER (Primary Key, Auto-increment)
- scheme_name: VARCHAR(100) (Unique, Not NULL)
- description: TEXT (Not NULL)
- budget: NUMERIC(15,2) (Not NULL, must be $\geqslant 0$)
- overseen_by: INTEGER (Not NULL, Foreign Key)

Functionalities Implemented

User Authentication & Access Control

- Secure login/logout system for different user roles (Admin, Employees, Citizens, Government Monitors).
- Role-based access control to restrict functionalities based on user type.

User Registration & Management

- Citizens and Panchayat employees can register using an online form.
- Passwords are stored securely as hashes for enhanced security.

Citizen Information Access

- Citizens can view panchayat-related information such as agriculture, health, education, welfare schemes, and infrastructure.
- Search functionality for quick access to relevant data.

Panchayat Employee Portal

- Employees can create and delete village records.
- Facilities for entering new schemes, infrastructure projects, and welfare initiatives.

Government Monitoring & Reports

- Government monitors can generate and download reports on village statistics.
- Reports on agriculture, health, and education sectors.

Database Management (Create & Delete Operations)

- Authorized employees can create new records and delete existing ones.
- Data stored for citizens, employees, welfare schemes, and infrastructure projects.

Secure & User-Friendly Dashboard

- Separate dashboards for Panchayat employees, citizens, and government agencies.
- Menu-based navigation for ease of access.

Mobile & Responsive Design

- Web application optimized for mobile devices.
- Responsive UI for different screen sizes.

AI-Powered SQL Query Generation

- Integrated Together AI to convert natural language queries into SQL.
- The system reads the database schema from a file and dynamically generates SQL queries based on user input.

Frontend Tools

Frontend
next: React framework for server-side rendering and static site generation.
react: Core library for building UI components.
react-dom: Provides DOM-specific methods for React applications.
tailwindcss: Utility-first CSS framework for designing UI components.
shadcn/ui: Pre-built UI components for React, based on Radix UI and Tailwind CSS.
framer-motion: Animation library for React applications.
lucide-react: Icon library for React applications.
sonner: Lightweight notification library for React.
zustand: State management library for managing global state.

Backend Tools

Backend

- pg: PostgreSQL client for Node.js to interact with the database.
- axios: Promise-based HTTP client for making API requests.
- bcrypt: Library for hashing passwords securely.
- jsonwebtoken: Library for handling authentication using JWT.
- dotenv: Loads environment variables from a .env file.
- zod: Schema validation library for ensuring correct data structures.

Development Tools

Development Tools

- typescript: Superset of JavaScript adding static typing.
- postcss: Tool for transforming CSS with plugins.