

DBMS Assignment 4 Report

Team Name : ACHA

Team Members:

- 1) Harshit Singh (22CS10031)
- 2) Aryan Sanghi (22CS30013)
- 3) Chandransh Singh (22CS30017)
- 4) Arpit Sodhani (22CS30012)

ER Diagram:

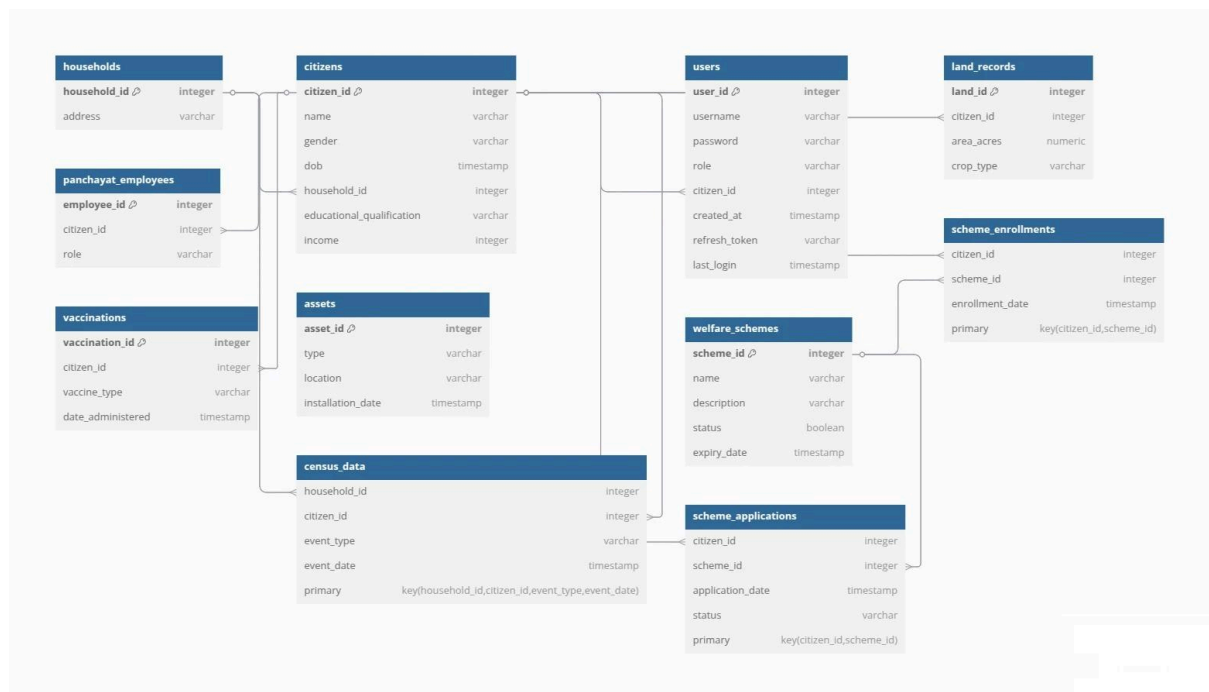


Table Schema and Relations :

Table households {
household_id integer [primary key]
address varchar
}

Table citizens {
citizen_id integer [primary key]
name varchar
gender varchar
dob timestamp

```
household_id integer
educational_qualification varchar
income integer
}
```

```
Table users {
  user_id integer [primary key]
  username varchar [unique]
  password varchar
  role varchar [note: 'admin, employee, citizen, monitor']
  citizen_id integer
  created_at timestamp
  refresh_token varchar
  last_login timestamp
}
```

```
Table land_records {
  land_id integer [primary key]
  citizen_id integer
  area_acres numeric
  crop_type varchar
}
```

```
Table panchayat_employees {
  employee_id integer [primary key]
  citizen_id integer
  role varchar
}
```

```
Table assets {
  asset_id integer [primary key]
  type varchar
  location varchar
  installation_date timestamp
}
```

```
Table welfare_schemes {
  scheme_id integer [primary key]
  name varchar
  description varchar
  status boolean
  expiry_date timestamp
}
```

```
Table scheme_enrollments {
  citizen_id integer,
  scheme_id integer,
  enrollment_date timestamp,
```

```
primary key (citizen_id, scheme_id)
}
```

```
Table vaccinations {
  vaccination_id integer [primary key]
  citizen_id integer
  vaccine_type varchar
  date_administered timestamp
}
```

```
Table census_data {
  household_id integer,
  citizen_id integer,
  event_type varchar,
  event_date timestamp,
  primary key (household_id, citizen_id, event_type, event_date)
}
```

```
Table scheme_applications {
  citizen_id integer,
  scheme_id integer,
  application_date timestamp,
  status varchar,
  primary key (citizen_id, scheme_id)
}
```

Relation: citizens.household_id > households.household_id // many-to-one
Relation: users.citizen_id > citizens.citizen_id // many-to-one
Relation: land_records.citizen_id > citizens.citizen_id // many-to-one
Relation: panchayat_employees.citizen_id > citizens.citizen_id // many-to-one
Relation: scheme_enrollments.citizen_id > citizens.citizen_id // many-to-one
Relation: scheme_enrollments.scheme_id > welfare_schemes.scheme_id // many-to-one
Relation: vaccinations.citizen_id > citizens.citizen_id // many-to-one
Relation: census_data.household_id > households.household_id // many-to-one
Relation: census_data.citizen_id > citizens.citizen_id // many-to-one
Relation: scheme_applications.citizen_id > citizens.citizen_id // many-to-one
Relation: scheme_applications.scheme_id > welfare_schemes.scheme_id // many-to-one

List of Functionalities Implemented:

Citizen Functionalities:

- View active welfare schemes
- List all available welfare schemes
- Track their own scheme applications
- View their vaccination history
- View their household details
- View Panchayat employees

- View panchayat-owned assets
- View census data
- View land records

Admin Functionalities:

- Add new users
- Remove users

Panchayat Employee Functionalities:

- Manage citizen records and profiles
- Process applications and manage welfare scheme enrollments
- Record births, deaths, and migrations in census management
- Manage land ownership and usage records
- Schedule and record vaccinations
- Update and maintain panchayat assets

Government Monitor Functionalities:

- View and report village statistics
- Manage vaccination records
- Update and view census data
- Monitor panchayat assets
- Verify land records

List of tools used :

Backend -

- Framework & Server:
 - Express.js : Handles backend routing and API endpoints.
- Database & ORM:
 - PostgreSQL : Database driver for connecting to PostgreSQL.
- Authentication & Security:
 - JWT : Manages authentication via JSON Web Tokens.
 - Bcrypt : Hashes and secures user passwords.
 - CORS : Enables cross-origin requests.
 - Cookie Parser : Handles cookies for authentication.
- Environment & Development Tools:
 - Dotenv : Manages environment variables.
 - Nodemon : Automatically restarts the server on changes.

Frontend -

- Framework & UI Library:

- React.js : Frontend framework for building UI.
 - React Router : Handles client-side navigation.
 - Material UI : Component library for styling.
 - Emotion : CSS-in-JS styling solution.
- Icons & Fonts:
 - FontAwesome : Provides icons.
 - React Icons : Another icon library.
- HTTP Requests & API Handling:
 - Axios : Handles API requests.
- Testing:
 - Testing Library : Used for unit and UI testing.
- Performance & Optimization:
 - Web Vitals : Monitors app performance.

Tech Stack Summary

- Frontend: React.js, React Router, Material UI, Emotion, FontAwesome, Axios.
- Backend: Node.js, Express.js, PostgreSQL, JWT, Bcrypt, Dotenv.
- Testing: React Testing Library.