ජnapada

Residential Management

Grama Niladhari Division Sri Lanka

Comprehensive User Guide & Technical Document.

E2320154
Mohamed Nizlan

Table of Contents

1. Introduction

The Resident Management System is designed to efficiently store and manage resident information for municipal and community administration purposes. This guide provides comprehensive instructions for database setup, configuration, usage, and troubleshooting

1.1 System Requirements

- **Operating System**: Windows 10/11, macOS 12+, or Linux (Ubuntu 20.04+)
- **MySQL Server**: Version 5.7 or later
- **PHP**: Version 7.4 or newer
- **Web Server**: Apache 2.4+/Nginx 1.18+ or XAMPP/WAMP package
- **Web Browser**: Chrome, Firefox, Edge, or Safari (latest versions)

2. Database Setup Instructions

2.1 Prerequisites

Before installation, ensure you have:

- **MySQL Server** installed and running
- **PHP** with MySQL extensions enabled
- **Web Server** configured and accessible
- **Administrative privileges** for database creation

2.2 Database Creation

1. Open **MySQL Command Line** or **phpMyAdmin**.
2. Execute the following SQL commands:

```
-- Create the database
CREATE DATABASE resident_database;

-- Select the database
USE resident_database;

-- Create residents table
CREATE TABLE residents (
    id INT AUTO_INCREMENT PRIMARY KEY,
    full_name VARCHAR(100) NOT NULL,
    dob DATE NOT NULL,
    nic VARCHAR(12) UNIQUE NOT NULL,
    phone VARCHAR(15) NOT NULL,
    address TEXT NOT NULL,
    email VARCHAR(100) NOT NULL,
    occupation VARCHAR(50),
    gender ENUM('Male', 'Female', 'Other') NOT NULL,
    registered_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

```
-- Create activity log table
CREATE TABLE activity_log (
   log_id INT AUTO_INCREMENT PRIMARY KEY,
   user_id INT NOT NULL,
   action_type ENUM('INSERT', 'UPDATE', 'DELETE', 'VIEW') NOT NULL,
   record_id INT,
   details TEXT,
   timestamp TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Create users table for authentication
CREATE TABLE users (
   user_id INT AUTO_INCREMENT PRIMARY KEY,
   username VARCHAR(50) UNIQUE NOT NULL,
   password VARCHAR(255) NOT NULL,
   full_name VARCHAR(100) NOT NULL,
   role ENUM('admin', 'officer', 'viewer') NOT NULL,
   last_login TIMESTAMP
);

-- Insert default admin user
INSERT INTO users (username, password, full_name, role) VALUES
('admin', '$2y$10$qNCLcGs/qP8mZA9SzlOMDu5VPnDCZYQKtEqo2iYzZFuRix5TwT4vG', 'System
Administrator', 'admin');
-- Note: Default password is 'admin123' (hashed)
```

2.3 Database User Configuration

For enhanced security, create a dedicated database user:

```
CREATE USER 'grama_user'@'localhost' IDENTIFIED BY 'SecurePass123!';
GRANT ALL PRIVILEGES ON resident_database.* TO 'grama_user'@'localhost';
FLUSH PRIVILEGES;
```

2.4 Application Configuration

1. Locate the config directory in your application installation folder
2. Edit connect.php with your database credentials:

```
<?php
define('DB_HOST', 'localhost');
define('DB_USER', 'grama_user');  // or 'root' if no separate user
define('DB_PASS', 'SecurePass123!');
define('DB_NAME', 'resident_database');
define('DB_PORT', 3306);
// Optional: Configure connection settings
$options = [
   PDO::ATTR_ERRMODE            => PDO::ERRMODE_EXCEPTION,
   PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC,
   PDO::ATTR_EMULATE_PREPARES   => false,
];
// Test connection
try {
   $pdo = new PDO("mysql:host=".DB_HOST.";dbname=".DB_NAME.";port=".DB_PORT, DB_USER,
DB_PASS, $options);
   // Connection successful
} catch (PDOException $e) {
   // Connection failed
   die("ERROR: Could not connect. " . $e->getMessage());
}?>
```

3. Using the Resident Management System

3.1 Logging In

1. Open the application in a web browser (e.g., http://localhost/residents/login.php).
2. Enter the default credentials:
   - **Username:** admin
   - **Password:** admin123
3. Click **Login** to access the dashboard.

3.2 Adding a New Resident

1. **Navigate to "Add Resident"**
   - Click **"Add Resident"** in the sidebar or the **"Add Resident"** button on the dashboard.
2. **Fill in Resident Details**
   - **Full Name** (Required)
   - **Date of Birth** (Required, format: YYYY-MM-DD)
   - **NIC Number** (Required, format: 123456789V or 123456789012)
   - **Phone Number** (Required, 10 digits, e.g., 0712345678)
   - **Address** (Required)
   - **Email** (Required, must be valid)
   - **Occupation** (Optional)
   - **Gender** (Required: Male/Female/Other)
3. **Submit the Form**
   - Click **"Register Resident"** to save the record.
   - If validation fails, errors will be displayed.

3.3 Searching for Residents

1. **Access the Search Feature**
   - From the **Residents List**, use the search bar at the top.
2. **Choose Search Method**
   - **By Name:** Partial matches allowed (e.g., "John" finds "John Silva").
   - **By Address:** Searches for any part of the address (e.g., "Colombo").
   - **By NIC:** Exact match required (e.g., "851234567V").
3. **View Results**
   - Matching records appear in a table.
   - Click **"View" (👁️)** to see full details.

3.4 Editing Resident Records

1. **Find the Resident**
   - Search for the resident using the methods above.
2. **Open the Edit Form**
   - Click the **"Edit" (✏️)** button next to the resident's name.
3. **Modify Details**
   - Update any fields (NIC must remain unique).
   - Click **"Update Resident"** to save changes.

3.5 Deleting a Resident Record

1. **Locate the Resident**
   - Search for the resident in the list.
2. **Initiate Deletion**
   - Click the **"Delete" (🗑️)** button.
3. **Confirm Deletion**
   - A confirmation page will appear.
   - Click **"Confirm Delete"** to permanently remove the record.

3.6 Logging Out

- Click **"Logout"** in the sidebar to securely exit the system.

Summary of User Actions

Action	Steps
Add Resident	Fill form → Submit
Search Resident	Use search bar → View results
Edit Resident	Find → Edit → Save changes
Delete Resident	Find → Delete → Confirm
Logout	Click "Logout"

4. Database Usage

4.1 Inserting Sample Data

Run this SQL to populate test records:

```
INSERT INTO residents (full_name, dob, nic, phone, address, email, occupation, gender) VALUES
('John Silva', '1985-05-15', '851234567V', '0712345678', '123 Main St, Colombo', 'john@email.com',
'Teacher', 'Male'),
('Nimali Perera', '1990-08-20', '901234568V', '0771234567', '456 Galle Rd, Galle', 'nimali@email.com',
'Doctor', 'Female'),
('Ravi Gunawardena', '1975-12-03', '751234569V', '0761234567', '789 Kandy Rd, Kandy', 'ravi@email.com',
'Engineer', 'Male'),
('Sarah Fernando', '1995-03-10', '951234570V', '0721234567', '234 Beach Rd, Negombo',
'sarah@email.com', 'Accountant', 'Female');
```

4.2 Common Database Queries

Below are examples of frequently used queries:

Search for Resident by NIC:

```
SELECT * FROM residents WHERE nic = '851234567V';
```

List Residents by Area (Address Contains):

```
SELECT * FROM residents WHERE address LIKE '%Colombo%';
```

Update Resident Details:

```
UPDATE residents SET phone = '0712345679', email = 'john.silva@email.com' WHERE id = 1;
```

4.3 Backup and Restore

Creating Database Backup:

```
mysqldump -u grama_user -p resident_database > resident_backup_[DATE].sql
```

Restoring from Backup:

```
mysql -u grama_user -p resident_database < resident_backup_[DATE].sql
```

Scheduled Backups (Linux Cron example):

```
# Add to crontab for daily backups at 2AM
0 2 * * * mysqldump -u grama_user -p'SecurePass123!' resident_database >
/path/to/backups/resident_backup_$(date +\%Y\%m\%d).sql
```

5. Advanced Configuration

5.1 Database Indexing

For improved query performance, add the following indexes:

```
-- Index for faster name searches
CREATE INDEX idx_name ON residents(full_name);

-- Index for faster address searches
CREATE INDEX idx_address ON residents(address(100));

-- Index for faster date-based queries
CREATE INDEX idx_registered_date ON residents(registered_date);
```

5.2 Foreign Key Relationships

If implementing additional tables, use foreign keys for data integrity:

```
-- Create a family unit table example
CREATE TABLE family_units (
   family_id INT AUTO_INCREMENT PRIMARY KEY,
   family_name VARCHAR(100),
   address TEXT,
   registered_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);

-- Add family_id to residents table
ALTER TABLE residents ADD COLUMN family_id INT;
ALTER TABLE residents ADD CONSTRAINT fk_family
   FOREIGN KEY (family_id) REFERENCES family_units(family_id);
```

6. Troubleshooting Guide

6.1 Common Issues and Solutions

Issue	Resolution
Login failures	• Check username and password • Ensure caps lock is off • Reset password if necessary
Connection errors	• Verify connect.php credentials • Check MySQL service status • Ensure port 3306 is open • Test connection with command line client
NIC format rejection	• Use 123456789V (old format) • Use 123456789012 (new format) • Check for leading/trailing spaces
Duplicate NIC entries	• NIC must be unique; check existing data • Use SELECT * FROM residents WHERE nic = '[NIC]' to find duplicates
Date format errors	• Strictly use YYYY-MM-DD format (e.g., 1995-11-25) • Check system locale settings
Slow loading pages	• Ensure proper database indexing • Check server resources • Optimize PHP configuration
Search not working	• Verify search term format • Check if records exist • Try alternative search criteria

6.2 Error Logging

Database errors are logged in:

- MySQL error log: /var/log/mysql/error.log (Linux) or C:\ProgramData\MySQL\MySQL Server\Data\[hostname].err (Windows)
- Application log: [app_directory]/logs/database_errors.log

Review these logs when troubleshooting persistent issues.

7. Maintenance and Security

7.1 Regular Maintenance Tasks

- **Daily**: Review system logs
- **Weekly**: Review error logs
- **Monthly**: Optimize tables using `OPTIMIZE TABLE residents;`
- **Quarterly**: Review user permissions and update passwords

7.2 Security Best Practices

- Change the default admin password immediately after installation
- Use strong passwords with mixed case, numbers, and symbols
- Enable password rotation every 90 days
- Limit MySQL server access to localhost if possible
- Enable SSL for remote database connections
- Implement input validation to prevent SQL injection
- Set up system backups on a regular schedule

8. Appendix

8.1 Database Schema Diagram

Table: residents

Field	Type	Null	Key	Default
id	INT	NO	PRI	AUTO_INCREMENT
full_name	VARCHAR(100)	NO		NULL
dob	DATE	NO		NULL
nic	VARCHAR(12)	NO	UNI	NULL
phone	VARCHAR(15)	NO		NULL
address	TEXT	NO		NULL
email	VARCHAR(100)	NO		NULL
occupation	VARCHAR(50)	YES		NULL
gender	ENUM	NO		NULL
registered_date	TIMESTAMP	NO		CURRENT_TIMESTAMP

Table: users

Field	Type	Null	Key	Default
user_id	INT	NO	PRI	AUTO_INCREMENT
username	VARCHAR(50)	NO	UNI	NULL
password	VARCHAR(255)	NO		NULL
full_name	VARCHAR(100)	NO		NULL
role	ENUM	NO		NULL
last_login	TIMESTAMP	YES		NULL

Table: activity_log

Field	Type	Null	Key	Default
log_id	INT	NO	PRI	AUTO_INCREMENT
user_id	INT	NO		NULL
action_type	ENUM	NO		NULL
record_id	INT	YES		NULL
details	TEXT	YES		NULL
timestamp	TIMESTAMP	NO		CURRENT_TIMESTAMP

8.2 Version History

Version	Date	Changes
1.0	29th of March, 2025	Initial release
0.9	01st of April, 2025	Beta testing version
0.8	08th of April, 2025	User interface improvements
0.7	11th of Apreil, 2025	Core functionality implementation

8.3 Support Contact Information

For technical assistance, please contact:

- **Email**: e2320154@bit.uom.lk
- **Phone**: +94 75 3627840
- **Hours**: Monday-Friday, 8:30 AM - 5:00 PM