

city_resident_id_card_management_system/: Root directory for the project.

- |—— admin/: Directory for admin-related files.
 - | |—— admin_dashboard.php: Dashboard for admin functionalities.
 - | | - This PHP file contains the dashboard interface for admin users, providing an overview of system functionalities and administrative controls.
 - | |—— admin_register_resident.php: Page for admin to register residents.
 - | | - This PHP file allows admin users to register new residents by entering their personal information and relevant details.
 - | |—— admin_manage_users.php: Page for admin to manage user accounts.
 - | | - This PHP file enables admin users to manage user accounts, including adding, editing, or deleting user profiles.
 - | |—— admin_manage_components.php: Page for admin to manage city components (regions, zones, etc.).
 - | | - This PHP file provides admin users with functionalities to manage city components such as regions, zones, woredas, cities, and kebeles.
- |—— moderator/: Directory for moderator-related files.
 - | |—— moderator_dashboard.php: Dashboard for moderator functionalities.
 - | | - This PHP file contains the dashboard interface for moderator users, displaying relevant functionalities and controls for managing resident registrations.
 - | |—— moderator_manage_residents.php: Page for moderators to manage resident registrations.
 - | | - This PHP file allows moderators to view, review, and manage resident registrations, including approving or rejecting registration requests.
 - | |—— moderator_approve_resident.php: Page for moderators to approve resident registrations.
 - | | - This PHP file provides moderators with a form to review resident registration details and approve or reject them accordingly.
- |—— resident/: Directory for resident-related files.
 - | |—— resident_dashboard.php: Dashboard for resident functionalities.
 - | | - This PHP file contains the dashboard interface for resident users, displaying their ID card and relevant city information.
 - | |—— resident_register.php: Page for residents to register for ID cards.
 - | | - This PHP file allows residents to register for ID cards by providing their personal information, including image upload for ID card generation.

- |
 - |
 - |—— resident_view_id_card.php: Page for residents to view their ID cards.
 - |
 - | - This PHP file displays the ID card of the logged-in resident user, showing their personal information and photo.

- |
 - |—— includes/: Directory for common includes.

- |
 - |
 - |—— db_connection.php: PHP file for establishing database connection.

- |
 - |
 - | - This PHP script establishes a connection to the MySQL database, allowing other PHP files to interact with the database.

- |
 - |
 - |—— header.php: Header file to be included in all pages.

- |
 - |
 - | - This PHP file contains the header section of the website, including navigation menus and other common elements.

- |
 - |
 - |—— footer.php: Footer file to be included in all pages.

- |
 - |
 - | - This PHP file contains the footer section of the website, including copyright information and other common elements.

- |
 - |—— css/: Directory for CSS files.

- |
 - |
 - |—— style.css: Main stylesheet for the project.

- |
 - |
 - | - This CSS file contains styles for styling the HTML elements across all pages of the project.

- |
 - |—— js/: Directory for JavaScript files.

- |
 - |
 - |—— script.js: Main JavaScript file for the project.

- |
 - |
 - | - This JavaScript file contains client-side scripting for enhancing user interactions and functionalities.

- |
 - |—— images/: Directory for image files.

- |
 - | - This directory contains image files used in the project, such as user avatars, logos, or graphical elements.

- |
 - |—— index.php

- |
 - |—— login.php

- |
 - |—— logout.php

- |
 - |—— db/: Directory for database-related files.

- |
 - |
 - |—— city_resident_id_card_management.sql: SQL file for database setup and population.

- |
 - |
 - | - This SQL script contains the database schema and initial data required for setting up the City Resident ID Card Management System database.

-- Table: admin

```
CREATE TABLE IF NOT EXISTS admin (  
    admin_id INT AUTO_INCREMENT PRIMARY KEY,  
    username VARCHAR(50) NOT NULL UNIQUE,  
    password VARCHAR(255) NOT NULL  
);
```

-- Table: region

```
CREATE TABLE IF NOT EXISTS region (  
    region_id INT AUTO_INCREMENT PRIMARY KEY,  
    region_name VARCHAR(50) NOT NULL UNIQUE  
);
```

-- Table: zone

```
CREATE TABLE IF NOT EXISTS zone (  
    zone_id INT AUTO_INCREMENT PRIMARY KEY,  
    zone_name VARCHAR(50) NOT NULL,  
    region_id INT,  
    FOREIGN KEY (region_id) REFERENCES region(region_id) ON DELETE CASCADE  
);
```

-- Table: woreda

```
CREATE TABLE IF NOT EXISTS woreda (  
    woreda_id INT AUTO_INCREMENT PRIMARY KEY,  
    woreda_name VARCHAR(50) NOT NULL,  
    zone_id INT,  
    FOREIGN KEY (zone_id) REFERENCES zone(zone_id) ON DELETE CASCADE  
);
```

-- Table: city

```
CREATE TABLE IF NOT EXISTS city (  
    city_id INT AUTO_INCREMENT PRIMARY KEY,  
    city_name VARCHAR(50) NOT NULL UNIQUE,  
    woreda_id INT,  
    FOREIGN KEY (woreda_id) REFERENCES woreda(woreda_id) ON DELETE  
CASCADE  
);
```

-- Table: kebele

```
CREATE TABLE IF NOT EXISTS kebele (  
    kebele_id INT AUTO_INCREMENT PRIMARY KEY,  
    kebele_name VARCHAR(50) NOT NULL,  
    city_id INT,  
    FOREIGN KEY (city_id) REFERENCES city(city_id) ON DELETE CASCADE  
);
```

-- Table: moderator

```
CREATE TABLE IF NOT EXISTS moderator (  
    moderator_id INT AUTO_INCREMENT PRIMARY KEY,  
    username VARCHAR(50) NOT NULL UNIQUE,  
    password VARCHAR(255) NOT NULL,  
    kebele_id INT,  
    city_id INT, -- Added city_id for each moderator  
    FOREIGN KEY (kebele_id) REFERENCES kebele(kebele_id) ON DELETE CASCADE  
);
```

-- Table: residents

```
CREATE TABLE IF NOT EXISTS residents (  
    resident_id INT AUTO_INCREMENT PRIMARY KEY,  
    username VARCHAR(50) NOT NULL UNIQUE,  
    password VARCHAR(255) NOT NULL,  
    full_name VARCHAR(100) NOT NULL,  
    image VARCHAR(255),  
    region_id INT,  
    zone_id INT,  
    woreda_id INT,  
    city_id INT,  
    kebele_id INT,  
    status ENUM('pending', 'approved', 'disapproved') DEFAULT 'pending',  
    FOREIGN KEY (region_id) REFERENCES region(region_id) ON DELETE CASCADE,  
    FOREIGN KEY (zone_id) REFERENCES zone(zone_id) ON DELETE CASCADE,  
    FOREIGN KEY (woreda_id) REFERENCES woreda(woreda_id) ON DELETE  
CASCADE,  
    FOREIGN KEY (city_id) REFERENCES city(city_id) ON DELETE CASCADE,  
    FOREIGN KEY (kebele_id) REFERENCES kebele(kebele_id) ON DELETE CASCADE  
);
```

-- Table: events

```
CREATE TABLE IF NOT EXISTS events (  
    event_id INT AUTO_INCREMENT PRIMARY KEY,  
    event_name VARCHAR(100) NOT NULL,  
    event_description TEXT,  
    event_date DATE  
);
```