Project Overview

Project Name: Property Rental Management System (PRMS)

Project Overview

The goal of this project is to develop a web-based platform for managing rental properties. The system will streamline operations, improve rent collection, and enhance tenant management by providing a digital solution for tracking properties, agreements, payments, and reminders.

Business Context

Currently, the business operates using traditional methods for managing rental properties, which includes manual rent collection and tracking. This approach is inefficient and leads to issues such as late payments and lack of reminders. The new system aims to digitize these processes, allowing for better monitoring of rental agreements, payment schedules, and tenant interactions.

Target Users

1. **Admin:** Manages the overall system, including user roles, properties, and financial reports.
2. **Staff:** Handles day-to-day operations, including processing rental agreements and payments.
3. **Customers:** Can view available properties, make rental requests, and manage their accounts.

Objective

To develop a web portal that allows for the management of rental properties, tracking of rental agreements, tenant management, and financial transactions.

Key Features

1. **User Management**
   - User Registration/Login (Admin, Staff, Tenant)
   - Role-based access control
   - Profile management
2. **Inventory Management**
   - Add/Edit/Delete Equipment

- Categorization of Equipment (e.g., Cantering Plates, Concrete Mixture Machines, etc.)
- Track current stock levels
- Track equipment on rent
- Upload images and specifications for each item

3. **Rental Management**
   - Create Rental Agreements
   - Calculate rental fees based on duration and quantity
   - Manage advance deposits
   - Track outstanding payments
   - Generate invoices and receipts
4. **Customer Management**
   - Add/Edit/Delete Customer Profiles
   - Track rental history for each customer
   - View outstanding payments and rental agreements
5. **Reporting**
   - Generate reports on inventory status (available, rented, damaged)
   - Financial reports (total income, outstanding payments)
   - Customer reports (rental history, outstanding dues)
6. **Notifications**
   - Email/SMS notifications for due payments
   - Alerts for low stock levels
7. **Admin Dashboard**
   - Overview of current stock, rented items, and outstanding payments
   - Quick access to key functionalities

Technical Specifications

1. Technology Stack

- Frontend: HTML, CSS, JavaScript (React.js or Angular)
- Backend: Node.js or PHP (Laravel)
- Database: MySQL or PostgreSQL
- Hosting: Cloud-based (AWS, Azure, or DigitalOcean)

2. Security

- SSL encryption for data transmission
- Secure user authentication (JWT or OAuth)
- Regular backups of the database

3. Responsive Design

- Mobile-friendly interface
- Compatibility with major browsers

Project Structure

1. **User Interface (UI)**
   - Home Page
   - Login/Registration Page
   - Dashboard
   - Property Management Page
   - Rental Agreement Management Page
   - Payment Management Page
   - Tenant Management Page
   - Reports Page
   - Notifications Page

2. **Backend Structure**
   - Home Page
   - Login/Registration Page
   - Dashboard
   - Property Management Page
   - Rental Agreement Management Page
   - Payment Management Page
   - Tenant Management Page
   - Reports Page
   - Notifications Page

3. **Database Structure**
   - Users Table
     - UserID (Primary Key)
     - Username
     - Password (hashed)
     - Role (Admin, Staff, Tenant)
     - Contact Information
   - **Properties Table**
     - PropertyID (Primary Key)
     - Type (Flat, Room, Apartment)
     - Location
     - Rent Amount
     - Deposit Amount
     - Status (Available, Rented)
   - **Agreements Table**
     - AgreementID (Primary Key)
     - PropertyID (Foreign Key)

    - TenantID (Foreign Key)
    - Start Date
    - End Date
    - Rent Amount
    - Deposit Amount
    - Status (Active, Completed)
- **Payments Table**
    - PaymentID (Primary Key)
    - AgreementID (Foreign Key)
    - Amount
    - Payment Date
    - Status (Paid, Unpaid)
- **Tenants Table**
    - TenantID (Primary Key)
    - Name
    - Contact Information
    - Rental History

Implementation Plan

Phase 1: Requirement Gathering

- Discuss with stakeholders to finalize requirements.

Phase 2: Design

- Create wireframes and UI/UX designs.
- Design database schema.

Phase 3: Development

- Frontend and backend development.
- Integration of modules.

Phase 4: Testing

- Unit testing, integration testing, and user acceptance testing.

Phase 5: Deployment

- Deploy the application on a cloud server.
- Set up domain and SSL.

Phase 6: Training and Support

- Provide training to staff on how to use the system.
- Offer on-going support and maintenance.