Property Rental System Implementation Guide

This document provides a detailed implementation guide for the Property Rental System, a comprehensive web-based solution for property management and rental processes. The system is designed to handle the complete lifecycle of property rentals, from property listing to payment processing, with built-in security and user management.

System Requirements

Software Requirements: • Python 3.x • MySQL Server • Flask Framework • Required Python Packages: - Flask==2.0.1 - Flask-SQLAlchemy==2.5.1 - Flask-Login==0.5.0 - mysqlclient - Werkzeug==2.0.1 Hardware Requirements: • Minimum 4GB RAM • 1GB free disk space • Internet connection for web access

Database Schema

Table	Description	Key Fields
User	User accounts	id, username, email, password, role
Property	Property listings	property_id, landlord_id, address, type, status
Tenant	Tenant profiles	tenant_id, user_id, first_name, last_name
Lease	Lease agreements	lease_id, property_id, tenant_id, start_date, end_date
Payment	Payment records	payment_id, lease_id, amount, status
Notification	System notifications	notification_id, user_id, message, type

Implementation Details

1. User Authentication System • Implemented using Flask-Login • Secure password hashing with Werkzeug • Session management and protection • Role-based access control 2. Property Management Module • CRUD operations for properties • Property status tracking (Available/Rented) • Property search and filtering • Property details management 3. Lease Management System • Lease creation with validation • Lease status tracking • Lease history maintenance • Automatic property status updates 4. Payment Processing • Payment tracking system • Payment status updates • Payment history maintenance • Automatic notifications 5. Notification System • Real-time notifications • Email notifications • Status updates • User-specific notifications

API Endpoints

Endpoint	Method	Description
/register	POST	User registration
/login	POST	User authentication
/properties	GET	List properties
/property/add	POST	Add new property
/lease/add	POST	Create new lease
/payment/add	POST	Record payment
/notifications	GET	Get user notifications

Security Implementation

The system implements comprehensive security measures: 1. Authentication Security • Password hashing using Werkzeug's security functions • Session management with Flask-Login • Secure cookie handling • Protection against session hijacking 2. Data Security • SQL injection prevention through SQLAlchemy • Input validation and sanitization • XSS protection • CSRF protection 3. Access Control • Role-based access control (RBAC) • Permission-based authorization • Secure route protection • Resource access validation 4. Database Security • Prepared statements • Connection pooling • Secure database credentials • Transaction management