

PROJECT REPORT

1. INTRODUCTION

Project Overview

The Lease Management System (LMS) is developed to automate and streamline the management of rental agreements, tenant records, property listings, billing, and maintenance scheduling. As the demand for rental properties increases, property managers face growing complexity in handling leases efficiently. LMS addresses these needs through a centralized, user-friendly platform that improves operational productivity and tenant satisfaction.

Project Objectives

- Create a centralized platform for managing property leases.
- Automate tenant onboarding, lease tracking, and payment management.
- Provide tools for scheduling and tracking maintenance.
- Enable digital document storage (e.g., lease contracts).
- Generate real-time analytics and alerts for lease expiry and dues.

2. LITERATURE SURVEY

Existing Problems

- Manual lease handling prone to errors and delays.
- Inadequate tracking of lease terms, renewals, and payments.
- Fragmented maintenance records.
- Lack of transparency and communication with tenants.

References

1. Lee, M. (2021). Digital Tools in Property Management. Real Estate Journal.
2. Rao, V. (2020). Smart Leasing Systems. IEEE Publications.

3. PROBLEM STATEMENT

Traditional leasing practices suffer from inefficiencies due to paperwork, human error, and disconnected workflows. A digital Lease Management System is needed to automate lease lifecycles, centralize tenant records, and simplify property administration.

4. IDEATION & PROPOSED SOLUTION

Empathy Map Summary

Analyzed feedback from property managers and tenants to identify pain points in lease tracking, payment handling, and communication.

Key Ideas from Brainstorming

- Lease expiry reminders
- Automated rent invoices
- Online document storage
- Maintenance request tracking
- Tenant self-service portal

Finalized Solution

LMS will allow property managers to handle lease agreements, tenant profiles, maintenance logs, and payments through one integrated system.

5. REQUIREMENT ANALYSIS

Functional Requirements

- Register properties and tenants.
- Automate lease initiation, renewals, and terminations.
- Generate invoices and track payments.

- Handle maintenance requests and status.
- Store documents digitally.

Non-Functional Requirements

- Secure access and data encryption.
- User-friendly interface.
- Multi-property scalability.
- Reliable with automatic backups.
- Role-based access control.

6. PROJECT DESIGN

Data Flow Diagram

Depicts data flow between tenants, property managers, billing system, and maintenance module.

User Stories

Manager: Manage lease agreements.

Tenant: Receive reminders for rent and lease expiry.

Admin: Generate invoices and reports.

Technician: Manage maintenance tasks.

7. SOLUTION ARCHITECTURE

Components

- Frontend: HTML/CSS/JavaScript
- Backend: Python Flask
- Database: MySQL / SQLite
- Storage: Cloud-based document storage
- APIs: SMS/Email notifications

8. PROJECT PLANNING & SCHEDULING

Sprint Tasks

1. Setup environment
2. Lease and tenant logic
3. Maintenance and billing
4. Document uploads
5. Testing and deployment

9. CODING & FEATURES

Feature 1: Lease Lifecycle Management

Feature 2: Maintenance Scheduling

10. PERFORMANCE TESTING

- Load tested with 50 users
- Page load time < 2 seconds
- Passed 100+ unit tests

11. RESULTS

- Local server deployment
- Dashboard for leases and payments
- Verified email reminders

12. ADVANTAGES & DISADVANTAGES

Advantages

- Paperless process
- Transparency

- Streamlined operations

Disadvantages

- Initial training required
- Cloud integration costs

13. CONCLUSION

The Lease Management System simplifies leasing workflows, improves transparency, and provides better decision-making insights.

14. FUTURE SCOPE

- Mobile app
- Payment gateway integration
- AI-based lease forecasting
- Multi-location support

15. APPENDIX

- Source Code
- Database Schema
- UI Mockups
- Sample Lease Data
- Deployment Guide