**Property Management System Development Plan**

**Overview**

This plan outlines the stages and technologies involved in building a comprehensive property management system that supports property management, financial tracking, and personal expenditure management.

**Stage 1: Initial Setup and Core Technologies**

* **Objective:** Set up the development environment and establish the core architecture of the system.
* **Technologies:**
  + Frontend: HTML, CSS, JavaScript
  + Backend: Node.js with Express (considering your current tech stack, but Python with Flask/Django could be an alternative if preferred later)
  + Database: MySQL (initially, with an option to switch to PostgreSQL)
  + Version Control: Git for source code management
  + Deployment: Initial setup on a local server, later to be moved to a cloud provider like AWS or DigitalOcean

**Stage 2: Basic Property Management Features**

* **Objective:** Develop basic interfaces and backend functionality for property management.
* **Technologies:**
  + Frontend: Bootstrap for responsive design, jQuery for simpler DOM manipulations
  + Backend: RESTful APIs using Express to handle requests like adding, editing, and retrieving property data
  + Database: MySQL schemas for properties, stages, and documents
* **Features to Develop:**
  + Property listing page
  + Add/Edit property form
  + Property details page including document management

**Stage 3: Financial Management Implementation**

* **Objective:** Implement comprehensive financial tracking for properties, rentals, and rehabilitation projects.
* **Technologies:**
  + Frontend: Data visualization libraries (e.g., Chart.js for displaying financial metrics)
  + Backend: Expanded API endpoints for financial data handling
  + Database: Additional tables for loans, payments, expenses, and budgets
* **Features to Develop:**
  + Loan and payment tracking system
  + Expense reporting and categorization
  + Budget setup and tracking for property rehabilitation

**Stage 4: Rental and Tenant Management**

* **Objective:** Create functionalities for managing rentals and tenants.
* **Technologies:**
  + Frontend: Further use of Bootstrap and jQuery to handle forms and listings
  + Backend: Secure API endpoints for tenant data transactions
  + Database: Tables for rental income, tenant details, lease agreements
* **Features to Develop:**
  + Tenant management dashboard
  + Rental income and expenditure log
  + Lease agreement management

**Stage 5: Reporting and Tax Preparation**

* **Objective:** Develop reporting tools and export functionalities for tax preparation and management overview.
* **Technologies:**
  + Frontend: Libraries for generating PDFs or CSV files (e.g., jsPDF, PapaParse)
  + Backend: Complex querying capabilities for report generation
* **Features to Develop:**
  + Customizable financial reports
  + Export tools for financial data
  + Integration with accounting software (if feasible)

**Stage 6: Personal Expenditure Tracking and Final Integration**

* **Objective:** Finalize the system with personal expenditure tracking and ensure all components are integrated smoothly.
* **Technologies:**
  + Frontend: Additional UI components for personal finance management
  + Backend: Private APIs for handling personal data securely
* **Features to Develop:**
  + Personal dashboard for financial tracking
  + Alerts and reminders for personal financial management
  + Comprehensive testing and debugging

**Stage 7: Deployment and Maintenance**

* **Objective:** Deploy the application to a cloud server and enter the maintenance phase.
* **Technologies:**
  + Deployment: Docker for containerization, AWS or DigitalOcean for hosting
  + Continuous Integration/Continuous Deployment (CI/CD) pipelines to manage deployment and updates
* **Actions:**
  + Finalize deployment scripts and environment settings
  + Regular updates and patches
  + Ongoing support and feature enhancements based on user feedback