



Case Study

**Customised ERPNext Implementation
for Real Estate Company**

Case Study

Customised ERPNext Implementation for Real Estate Company.



Summary

We designed and implemented a custom ERPNext solution for a U.S.-based real estate client to streamline CRM, onboarding, listing management, and transaction workflows. This involved creating secure APIs, automating employee creation from leads, integrating Skyslope MLS listings, building custom DocTypes, and configuring email automation with dynamic content. The system enhanced real-time collaboration, increased onboarding speed, and introduced intelligent listing and task management capabilities.

Industry

Real Estate

Users/customers

Admin, Superadmin, Brokers, Agents, HR Recruiter

Tech Stack

Frappe, JavaScript, Python, AWS, MariaDB, CSS, Redis, Docker

Team Size

4 Oodlites

(1 Project Manager, 1 Backend Developer, 1 Devops, 1 QA Engineers)

Scope of Work (SoW)

The client's operations covered lead nurturing, HR onboarding, financial transactions, document storage, and multi-listing service (MLS) property management. Our task was to transform their loosely connected operations into a unified platform.

This required building:

1. APIs to automate CRM-to-Employee conversion
2. Custom DocTypes and child tables for managing onboarding documentation
3. Role-based access for different types of users (agents, admins, HR, brokers)
4. Real-time listing synchronization with Skyslope's MLS API
5. A secure webhook system to handle payment status updates

6. A dashboard system to link tasks, documents, and financial activity per employee
7. Integration with external tools such as Payload for transaction processing and document syncing
8. System-wide error handling, logging, and testing tools to ensure reliability across modules

Our development efforts were guided by the client's vision of automating their core business while preserving flexibility and user-friendliness.

Solution

We analyzed our client's requirements and had a kick-off meeting to put together a team of developers for the project. Our team began with a careful assessment of the client's existing business model and formulated an execution strategy to achieve the tasks at hand. We followed agile methodology to carry out the development process in a phased manner, starting with graphic design and then moving ahead with the frontend and backend interfaces. We also deployed 1 QA engineers to test each functionality against a given set of quality standards and one DevOps engineer (on-demand) on a time-and-material basis.

1. CRM to Employee Conversion API

We implemented a REST API in Frappe that transforms a CRM Lead into an Employee document. The system verifies whether the lead exists, fetches necessary data such as name, email, and contact information, and auto-creates a corresponding user in ERPNext. We embedded a "Convert to Employee" button directly into the Lead form, streamlining the entire onboarding pipeline for HR teams. This automation eliminated redundant entry work, reduced human error, and created a seamless transition from prospect to employee.

2. Custom Employee Doctype Enhancements

To align with the onboarding needs of the real estate domain, we extended the Employee Doctype by creating collapsible sections like "Employee Documents." These included fields for uploading Sign-Up PDFs, Real Estate Licenses, and Auto Insurance documents. We utilized Frappe's child table system to build structured and scalable fields, ensuring consistency and data cleanliness. The user interface was designed to be intuitive, enabling HR to manage required documentation without technical support.



3. Email Automation with Attachments

A key business requirement was to automate the welcome process. When an employee record was submitted with a document attached, a state-based filter checked eligibility for sending a welcome email. The emails included dynamically rendered employee data and attached documents in PDF format. We initially used `frappe.sendmail()` for simplicity but optimized it with `enqueue` to send messages asynchronously—ensuring high performance and fail-safe execution during large onboarding batches.

4. User Access Control and Role Profiles

To maintain data confidentiality and operational discipline, we built role-based permissions using ERPNext's Role Permission Manager. Each user was mapped to their own Employee record, and view/edit rights were restricted accordingly. We created user roles such as Agent, Broker, and HR with specific access to modules. Field-level control was applied where necessary, including sensitive fields like compensation and license information.

5. Global UI Improvements and Form Optimization

To improve usability and consistency, we introduced several UI enhancements across forms and dashboards. This included converting dropdowns to multi-select fields where multiple inputs were required, implementing section breaks to logically group fields, and introducing mandatory field indicators and placeholder help texts. These enhancements helped both admin users and agents navigate complex forms with ease and minimized input errors. Additionally, global loading indicators were added for better user experience during data-intensive operations like document uploads and listing imports.

6. Sales Invoice and Currency Handling

We built an API to automatically generate Sales Invoices in ERPNext, taking into account pricing rules and currency conversion. A major challenge was resolving currency mismatches—for instance, when the default Debtor account was in INR while the transaction was in USD. We updated account mappings and set default receivable accounts in company settings to ensure clean financial entries. Invoice PDFs were generated post-submission and automatically emailed to customers.



7. Secure Webhook for Payment Processing

The client needed a secure endpoint to process payment success events. We built a webhook in Frappe that validated requests using Basic Authentication. Credentials were securely stored in `site_config.json` and never hardcoded. On receiving a successful transaction status, the system auto-created a Payment Entry in ERPNext and logged the event for debugging. Using curl and Base64 headers, we extensively tested the endpoint for security and stability.

8. MLS Integration with Skyslope

To automate listing management, we integrated with Skyslope's `/files/listing` API. The response returned data in paginated format, requiring logic to iteratively collect all listing pages. Each field was mapped to our custom "Listing" Doctype. Duplicate prevention was a key feature—we checked whether a file ID already existed before insertion. Errors were gracefully logged using try-except blocks, ensuring smooth operations and reliable syncing.

9. Multi-Select Field Implementation

The real estate domain often requires associating employees with multiple licenses, languages, or tags. Since ERPNext does not support native multi-select links, we used child tables with link fields pointing to reference DocTypes. This design provided relational integrity while allowing users to select multiple values. It made data management modular, clear, and extensible.

10. Employee Dashboard and Task Sync

We enhanced the Employee dashboard to show CRM tasks assigned to a user. These tasks were fetched based on user ID mapping and displayed in a readable, structured table. Any HTML present in task descriptions was stripped for clarity. Additionally, any task updates from the CRM portal were reflected in near real-time in the employee's view, ensuring visibility and accountability.

11. Frappe CRM Lead Creation via Signup API

To automate lead capture from the client platform, we created a custom ERPNext API endpoint at `/api/method/your_app.api.create_lead_from_signup`. The endpoint accepted user data via POST (first name, last name, email, phone, company, etc.). The payload was mapped directly to the standard "CRM Lead" Doctype. Duplicate prevention was enforced by checking for existing email IDs before insertion. Data integrity was ensured via validation logic, and successful insertions triggered a commit to persist records in real time. The endpoint returned structured JSON responses for success and failure cases, enabling smooth frontend integration and error handling.

Results :

- The system went live after thorough testing and client review. The results were immediate and measurable:
- 70% reduction in manual onboarding time via CRM-Employee automation
- Improved accuracy in financial workflows through currency validation
- Secure payment tracking without manual reconciliation
- Real-time MLS listings synced to ERPNext without third-party tools
- High employee adoption due to intuitive document upload and dashboard views
- Bulletproof permissions system ensuring data visibility by role
- This implementation also laid the foundation for further expansion into customer support, listings syndication, and advanced analytics.

About Oodles ERP

Oodles ERP is a software services company that offers complete enterprise software development services with a focus on implementing next-gen technologies. With a proven track record in custom ERP development, we have successfully completed 50+ software projects related to CRM, HRM, inventory/warehouse, eCommerce, supply chain, and logistics. We are mainly focused on helping startups and small-to-medium enterprises to achieve digital transformation through cost-effective ERP software solutions.

Address

India Office:

Unit No. 110, IRIS Tech Park , Sector 48, Sohna Road, Gurgaon, India, 122018

US Office: 30N, Gloud St STR E, Sherdian, Wyoming (USA) - 82801

Singapore Office: 10 Anson Road, #13-09, International Plaza Singapore 079903

