SUBJECT OF THE INTERNSHIP REPORT: **Design and Implementation of a Multi-Tenant SaaS CRM Platform Using Odoo and Flutter Web**

Directed by: Maissa Drira

Supervised by: Ahmed Ben Miled

ESPRIT Supervisor: Hassen Jilani Ben Naouara Nouridine

Company Supervisor: Alwen.co

ABSTRACT

This report details my three-month internship with the Tunisian software company Alwen Co., where I helped create an enterprise-level CRM platform that was subscription-based. I obtained practical expertise in full-stack SaaS development, including frontend user experience design and backend architecture, while working in a cross-functional team. The internship gave me invaluable experience with contemporary software development techniques in a business setting, such as scalable web application architecture, multi-tenant system design, and API integration patterns. Through this training, I improved my analytical skills for assessing security needs and scalability issues in real-world systems, and I gained a broader understanding of the technical and business concerns involved in developing enterprise software solutions.

Keywords:

Odoo v18: Open-source ERP framework

Flutter: Cross-platform UI toolkit

JSON-RPC: Remote procedure call protocol

Riverpod: Flutter state management

SaaS: Software as a Service

ACKNOWLEDGEMENTS

I would want to express my gratitude to the Alwen Co. team for accepting me into their development group and providing me with the opportunity to work on an actual production project. I would want to express my appreciation to my supervisor and coworkers for their unwavering support, technical criticism, and cooperation during the internship. I also like to thank my academic supervisor for the helpful criticism that made my report better. Lastly, I want to express my gratitude to my friends and family for helping me through this fulfilling educational journey.

TABLE OF CONTENTS

LIST OF FIGURES

1 EXECUTIVE SUMMARY

This report details my three-month internship at Alwen Co., where I helped with the creation of a SaaS CRM platform for several tenants. Building backend modules in Odoo v18 for automatic CRM deployment and subscription management, as well as creating a Flutter web application with type-safe data processing, permission-based routing, and secure authentication, were the main goals of my work. My knowledge of SaaS design, backend-frontend integration, and performance optimization was enhanced by the internship, which also introduced me to scalability and security issues.

2 INTRODUCTION

I worked on a project at Alwen Co. during my internship that was designed to provide Tunisian companies with a customized SaaS CRM platform. Customers can access CRM functionality based on their subscription tier thanks to the platform's combination of an Odoo-based backend and a Flutter interface.

I was tasked with creating frontend components for routing, authentication, and data validation in addition to crucial backend modules for multi-tenant feature isolation and deployment automation. In order to ensure precise connection with Odoo models, I also contributed to user-facing modules like Sales, Projects, and Invoices.

The company's background, the extent of my internship, and the specifics of the work I did are all covered in this report. It also considers the difficulties I ran into, the compromises I had to make during deployment, and the knowledge I acquired. The purpose is to provide both an academic assessment of my learning and a professional account of my engineering contributions.

3 COMPANY CONTEXT

Description of the Company

Alwen Co. is a young startup specializing in the design of adaptable enterprise software and digital services. The company develops customizable tools that address the operational needs of small and medium-sized businesses, with a strong emphasis on flexibility and client-specific tailoring. Among its flagship initiatives is the 216.CBS project, a Flutter-based CRM frontend that integrates seamlessly with Odoo, enabling businesses to access a modern, user-friendly interface for their customer and operations management.

Mission and Objectives

Alwen's mission is to empower small and medium enterprises with adaptable, easy-to-deploy software solutions that streamline both customer relations and internal operations. For the 216.CBS project, the main objectives include: delivering a secure and cross-platform CRM frontend integrated with Odoo, enabling modular business features such as employee management, sales, leads, inventory, invoices, and expenses, and preparing the solution for multi-tenant SaaS deployment to support scalability and regional expansion

Industry Structure

Alwen positions itself within the enterprise software and SaaS ecosystem, a sector that prioritizes modular design, seamless integration, and secure multi-platform access. The competitive landscape includes specialized CRM/ERP vendors and system integrators offering localized solutions. Key drivers in this industry are rapid deployment with existing systems, robust data security, and lightweight, maintainable frontends that enhance user experience while lowering operational overhead.

Market Structure

Initially, Alwen focuses on the North African business ecosystem, starting with Tunisia, with an expansion roadmap targeting South African markets. The company leverages multiple distribution strategies, including direct sales to small and medium enterprises, partnerships with ERP implementation firms, and a planned online self-service subscription model. Its competitive edge lies in tight Odoo integration, a streamlined Flutter frontend, and a strong emphasis on configurable, secure deployments, all of which are key differentiators in a market where flexibility and trust are crucial.

4 INTERNSHIP DESCRIPTION

This section provides an overview of my internship experience, including the context of the placement, the objectives set, the challenges I faced, the specific responsibilities I undertook, and the key lessons learned. The main focus of my work was the development of a multi-tenant SaaS CRM platform, which required contributions to both the backend modules in Odoo v18 and the Flutter-based web application. My role involved ensuring secure subscription validation, automating CRM deployment, and implementing the architecture for the web app along with several features.

Internship Context

I completed my internship at Alwen Co., a startup based in Tunis that specializes in adaptable enterprise solutions. I worked alongside a development team of 3, and we worked actively and under tight deadlines. The project could be summarized into integrating Odoo as a back- end with a cross-platform Flutter frontend. I collaborated closely with my teammates on different parts of the system, participating in daily reviews and technical discussions. The company followed agile-inspired practices, with emphasis on version control, continuous documentation, and iterative progress tracking. My internship was carried out in a hybrid format, part of it on-site at the company's office and part of it remotely.

General and Specific Objectives of the Internship

The overall goal of my internship was to gain practical experience in SaaS software development and contributing meaningful features to the CRM system. More specifically, I aimed to strengthen my knowledge of optimizing frontend and backend communication for realtime software and practice my newly learned technology Flutter. I also set myself the objective of learning how to automate system deployments, since this was an essential requirement for scaling the CRM to multiple tenants.

Challenges and Obstacles

One of the main challenges I faced was the complexity of multi-tenant design. Isolating client data while ensuring efficient subscription validation required careful use of Odoo's permission model, especially when mapping features to groups and record rules. Odoo had very detailed documentation especially if you've paid permium prices, but we didn't, and contacting their support would get you far enough before they ask you to pay for a subscription. Security was another demanding area. I had to make sure that validation checks were consistently enforced server-side, as relying on client-side logic would have exposed the system to risks such as session hijacking or privilege escalation. Automating deployments was also not straightforward. Designing the CRM installer in a way that was both idempotent and capable of rolling back after failure forced me to consider dependencies between modules and their initialization sequences. On the frontend, I encountered difficulties with setting up the correct architecture. The routing system needed to load permissions before building accessible routes, which occasionally introduced latency that affected the user experience. I overcame these challenges by looking up discussions on these problems and how other experienced engineers approached them.

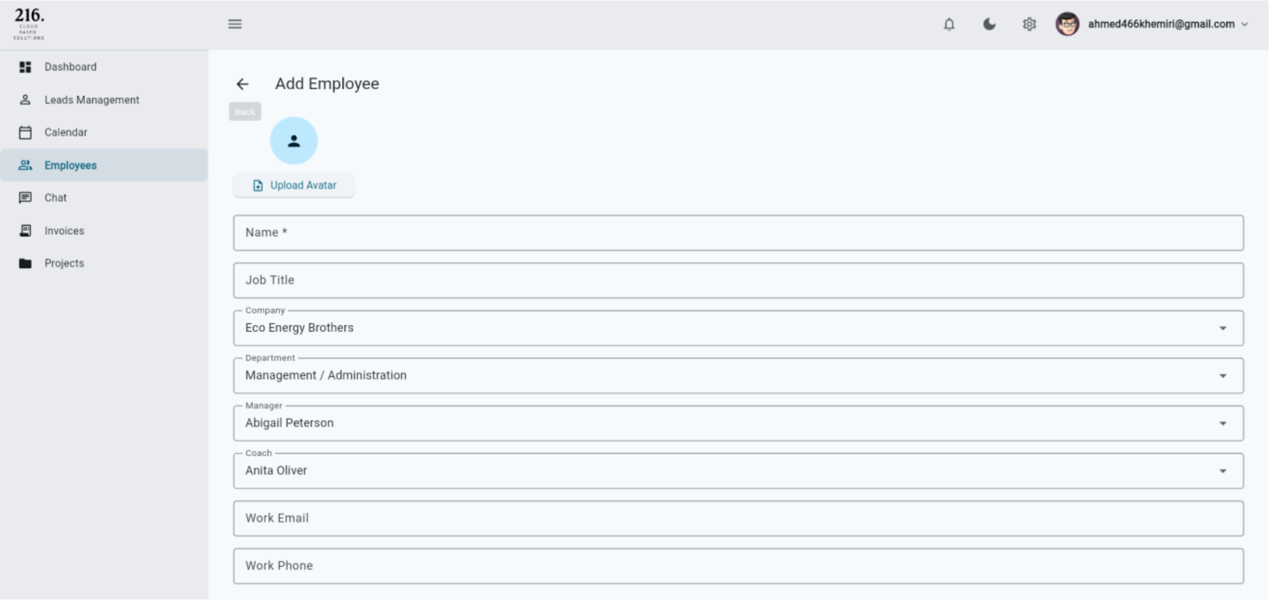
Assigned Tasks and Responsibilities

My responsibilities were split between backend and frontend development. On the backend, I implemented the Multi Client SaaS module, which handles subscription-based feature toggling and enforces tenant-level data isolation through Odoo's record rules and access rights. I also developed the 216.CRM Suite module, which automates the installation and configuration of dependent modules, and supports one-click deployment of CRM instances for new clients. In addition, I designed server-side validation endpoints to verify subscription status and enforce data integrity during write operations.

On the frontend, I've designed the architecture from the ground up, iterating over it through the weeks. Among the core systems I've built is the Odoo Service; This layer manages authentication, CRUD operations, error parsing, and retry mechanisms. I also built a permission-aware routing system using GoRouter in combination with Riverpod providers, which filters accessible routes dynamically according to the authenticated user's features. Another important responsibility was maintaining the field registry, which provides O(1) lookups for more than 200 fields and ensures that inputs are sanitized and type-checked before reaching the backend. I also worked on functional modules, including Sales, Projects, Invoices, Purchases, Employees, and Chat, implementing workflows such as record creation, detail navigation, and list filtering.

Key Learnings

The internship gave me significant exposure to both backend and frontend architecture. I learned how to design Odoo modules that enforce strict security policies, and I deepened my knowledge of record rules and multi-tenant data isolation. On the frontend, I became proficient in using Riverpod for state management and in designing routing systems that respect user permissions. Working on API communication taught me the importance of structured logging, error handling, and retry strategies. Just as importantly, I learned how to operate in a collaborative environment, where communication, documentation, and incremental delivery were essential to keep the project on track. Overall, the experience reinforced my understanding of scalability and developer productivity as central concerns in SaaS development.



Add Employee

Name *

Job Title

Company: Eco Energy Brothers

Department: Management / Administration

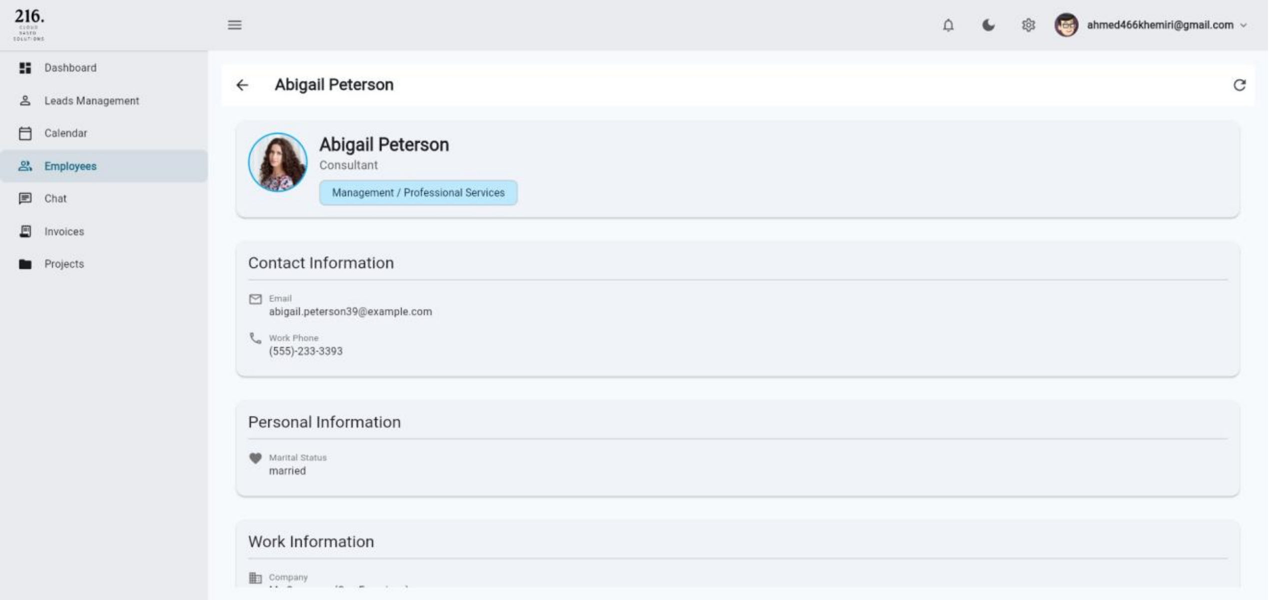
Manager: Abigail Peterson

Coach: Anita Oliver

Work Email

Work Phone

Figure 1: Employee creation form in Odoo, allowing HR managers to add new employees.



Abigail Peterson

Abigail Peterson
Consultant
Management / Professional Services

Contact Information

Email: abigail.peterson39@example.com

Work Phone: (555)-233-3393

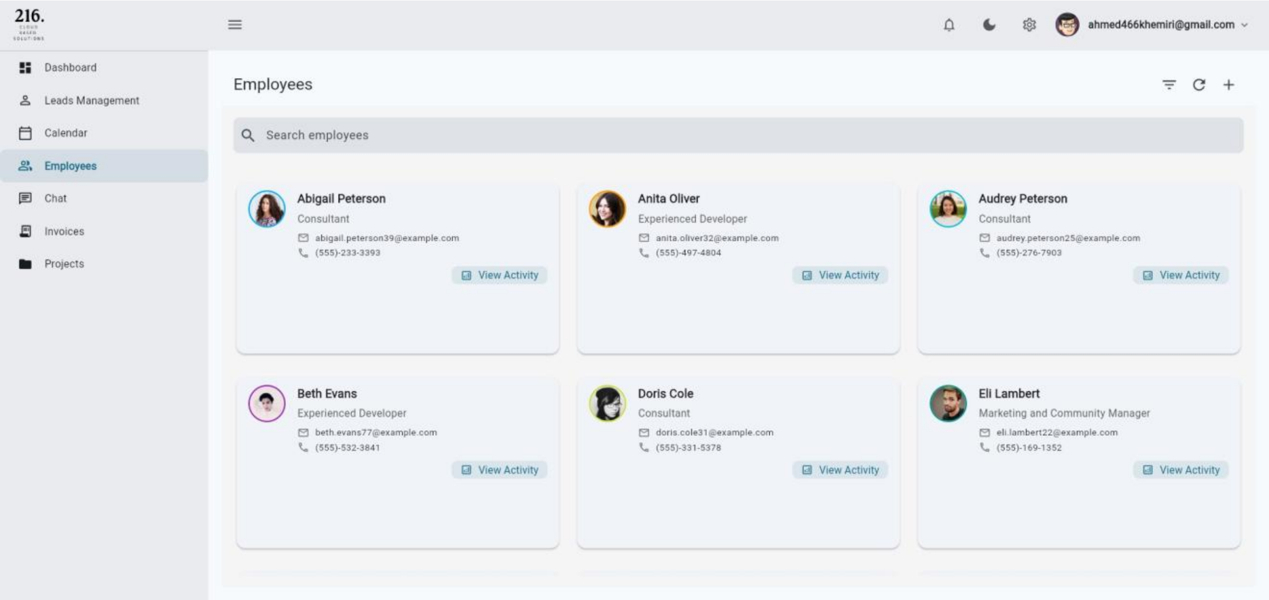
Personal Information

Marital Status: married

Work Information

Company

Figure 2: Employee details view, showing personal information and work-related data.



Employees

Search employees

Abigail Peterson
Consultant
abigail.peterson39@example.com
(555)-233-3393
View Activity

Anita Oliver
Experienced Developer
anita.oliver32@example.com
(555)-497-4804
View Activity

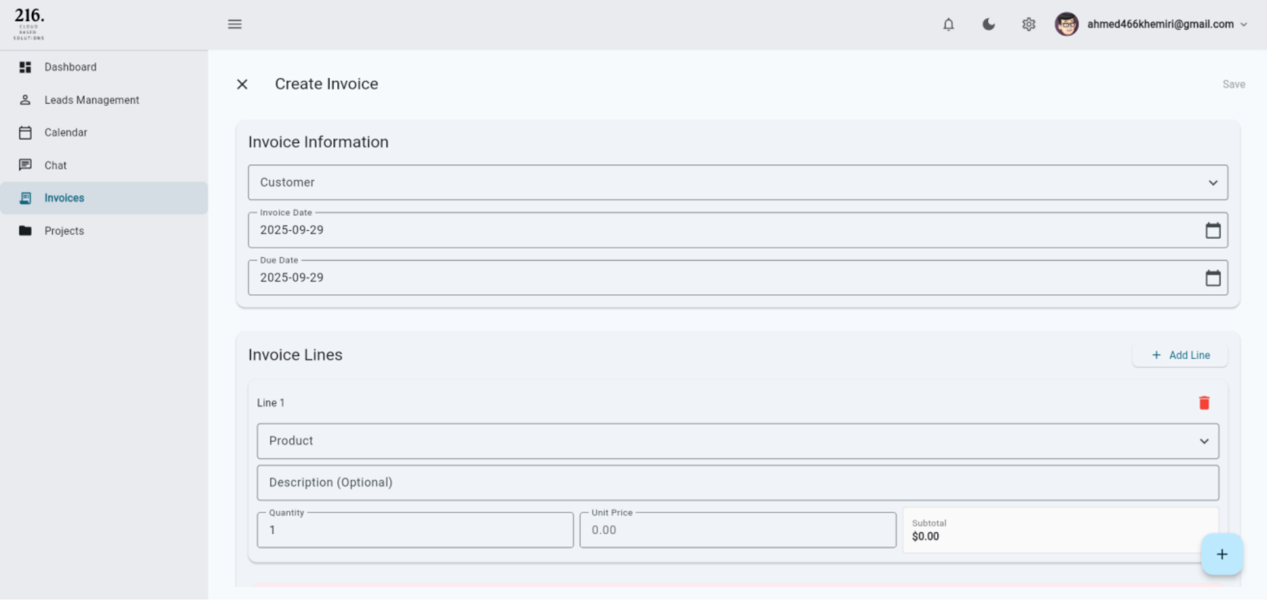
Audrey Peterson
Consultant
audrey.peterson25@example.com
(555)-276-7903
View Activity

Beth Evans
Experienced Developer
beth.evans77@example.com
(555)-532-3841
View Activity

Doris Cole
Consultant
doris.cole31@example.com
(555)-331-5378
View Activity

Eli Lambert
Marketing and Community Manager
eli.lambert22@example.com
(555)-169-1352
View Activity

Figure 3: Employee list interface displaying all registered employees in the system.



Create Invoice

Invoice Information

Customer

Invoice Date: 2025-09-29

Due Date: 2025-09-29

Invoice Lines

Line 1

Product

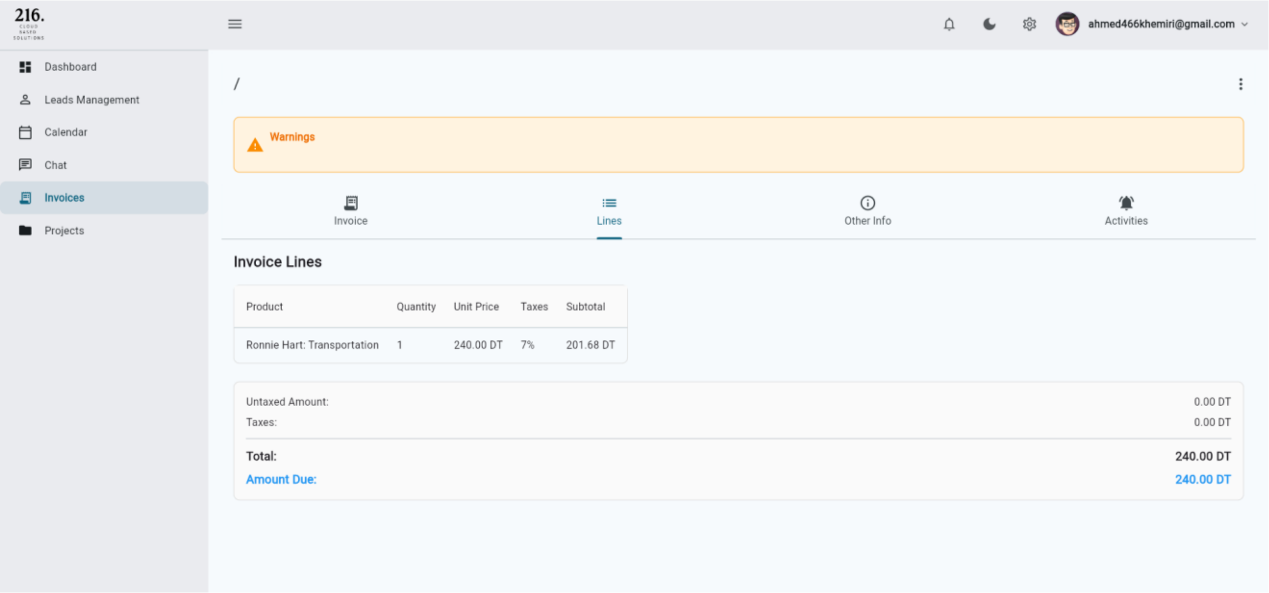
Description (Optional)

Quantity: 1

Unit Price: 0.00

Subtotal: $0.00

Figure 4: Invoice creation form where users can generate and configure a new invoice.

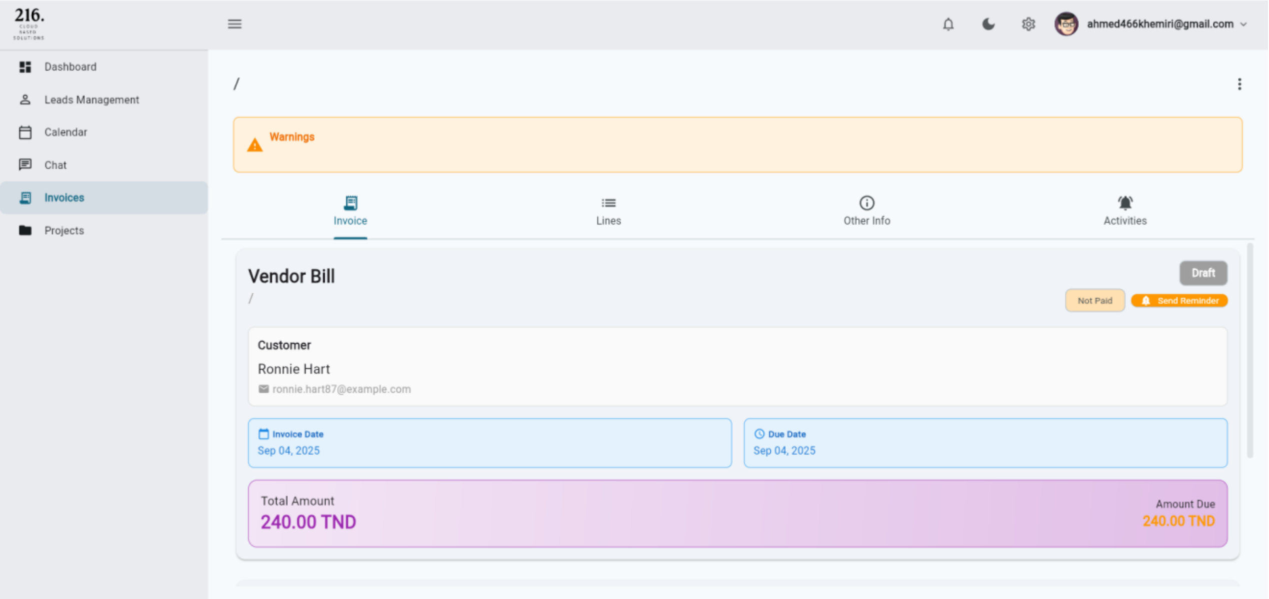


Invoice Lines

Product	Quantity	Unit Price	Taxes	Subtotal
Ronnie Hart: Transportation	1	240.00 DT	7%	201.68 DT

Untaxed Amount:	0.00 DT
Taxes:	0.00 DT
Total:	240.00 DT
Amount Due:	240.00 DT

Figure 5: Invoice details page showing the structure of a generated invoice (example 1).



Vendor Bill
/

Customer
Ronnie Hart
ronnie.hart87@example.com

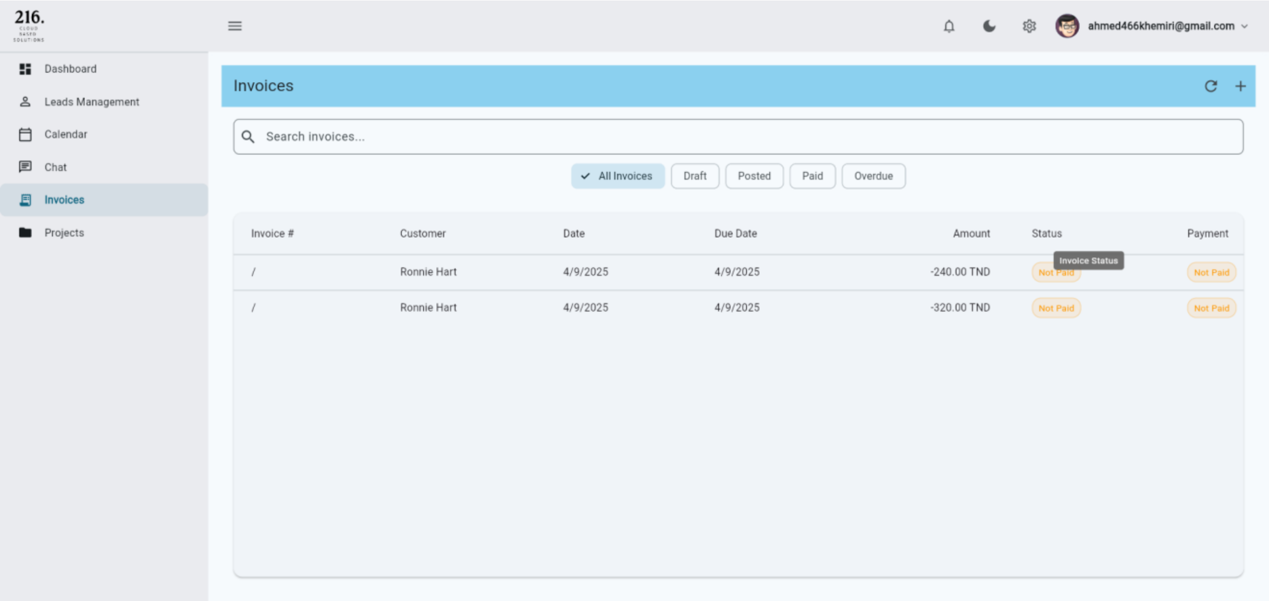
Invoice Date
Sep 04, 2025

Due Date
Sep 04, 2025

Total Amount
240.00 TND

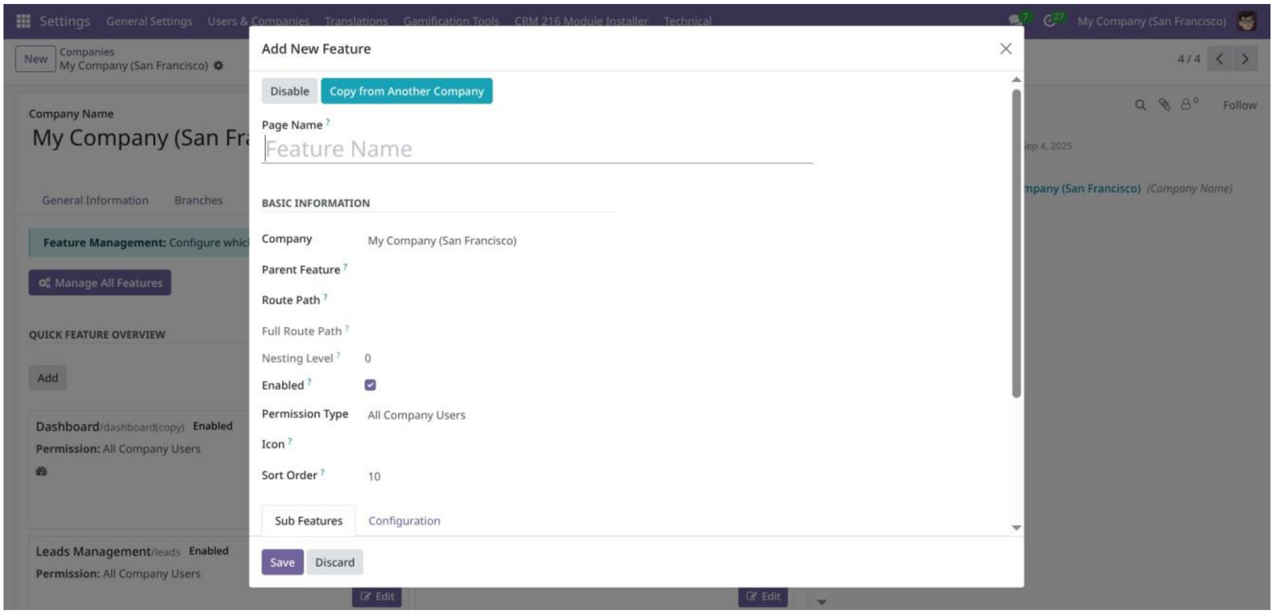
Amount Due
240.00 TND

Figure 6: Invoice details page with additional fields and line items (example 2).



Invoice #	Customer	Date	Due Date	Amount	Status	Payment
/	Ronnie Hart	4/9/2025	4/9/2025	-240.00 TND	Not Paid	Not Paid
/	Ronnie Hart	4/9/2025	4/9/2025	-320.00 TND	Not Paid	Not Paid

Figure 7: Invoice list view showing all generated invoices and their statuses.



Company Name
My Company (San Fra

Feature Management: Configure whic

Dashboard/dashboard(copy) Enabled
Permission: All Company Users

Leads Management/leads Enabled
Permission: All Company Users

ep 4, 2025

mpany (San Francisco) *(Company Name)*

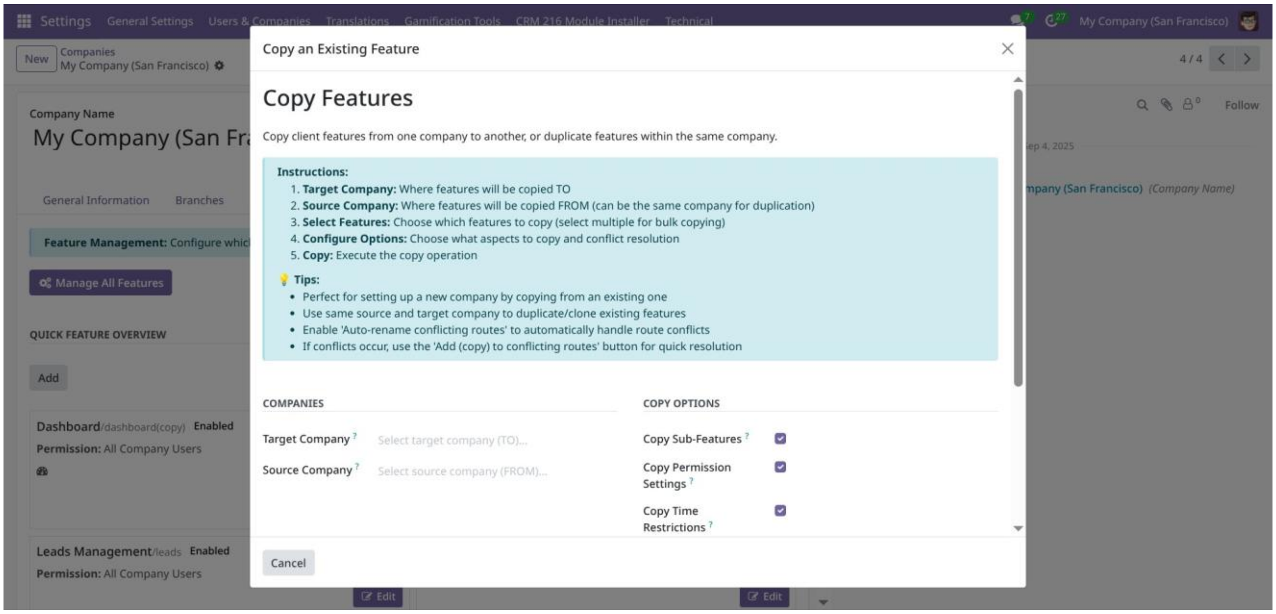
Add New Feature

Page Name: Feature Name

BASIC INFORMATION

Company	My Company (San Francisco)
Parent Feature	
Route Path	
Full Route Path	
Nesting Level	0
Enabled	☑
Permission Type	All Company Users
Icon	
Sort Order	10

Figure 8: Odoo features interface for creating new functionalities in the system.



Copy an Existing Feature

Copy Features

Copy client features from one company to another, or duplicate features within the same company.

Instructions:

1. **Target Company:** Where features will be copied TO
2. **Source Company:** Where features will be copied FROM (can be the same company for duplication)
3. **Select Features:** Choose which features to copy (select multiple for bulk copying)
4. **Configure Options:** Choose what aspects to copy and conflict resolution
5. **Copy:** Execute the copy operation

💡 **Tips:**

- Perfect for setting up a new company by copying from an existing one
- Use same source and target company to duplicate/clone existing features
- Enable 'Auto-rename conflicting routes' to automatically handle route conflicts
- If conflicts occur, use the 'Add (copy) to conflicting routes' button for quick resolution

COMPANIES

Target Company ? Select target company (TO)...

Source Company ? Select source company (FROM)...

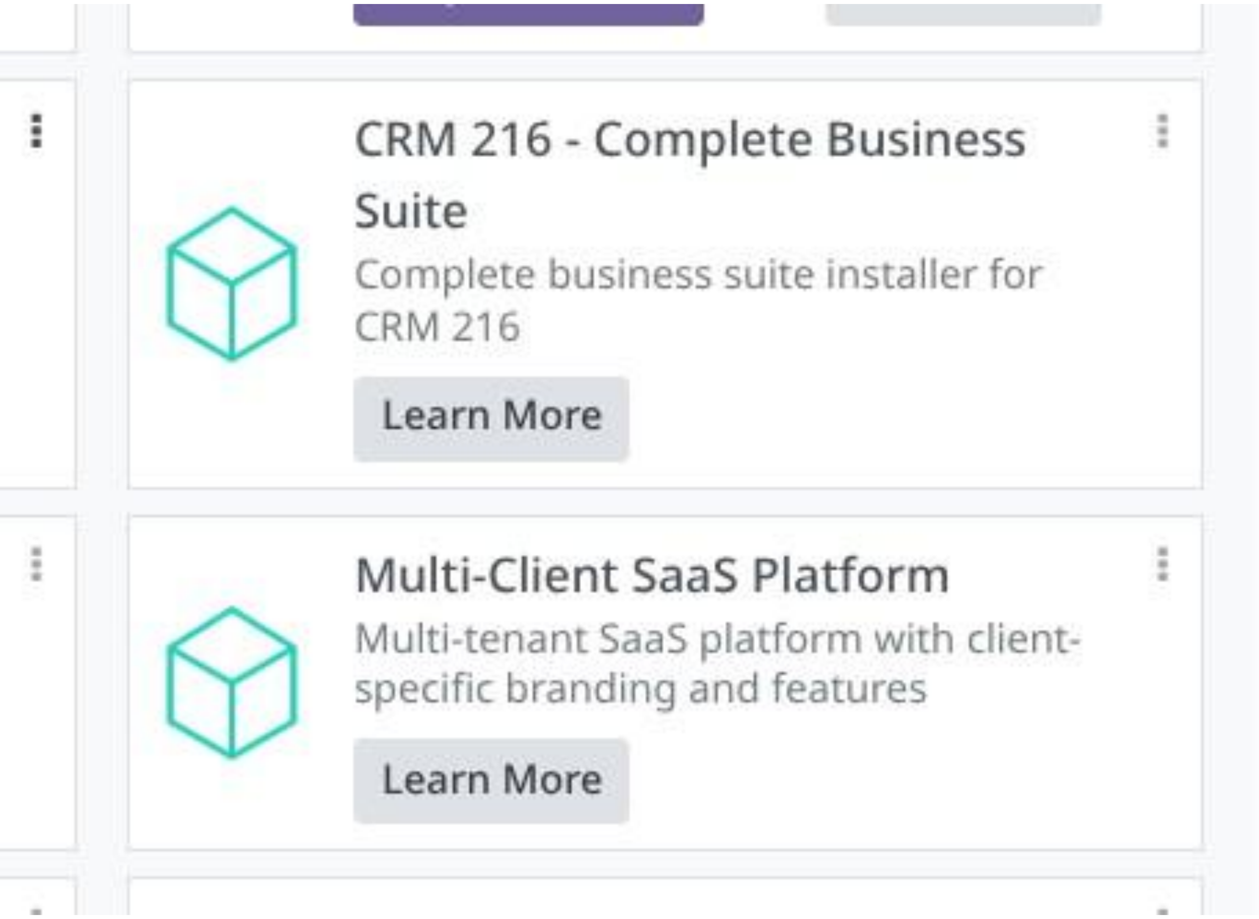
COPY OPTIONS

Copy Sub-Features ? ☑

Copy Permission Settings ? ☑

Copy Time Restrictions ? ☑

Figure 9: Odoo features replication screen used to duplicate existing modules or workflows.

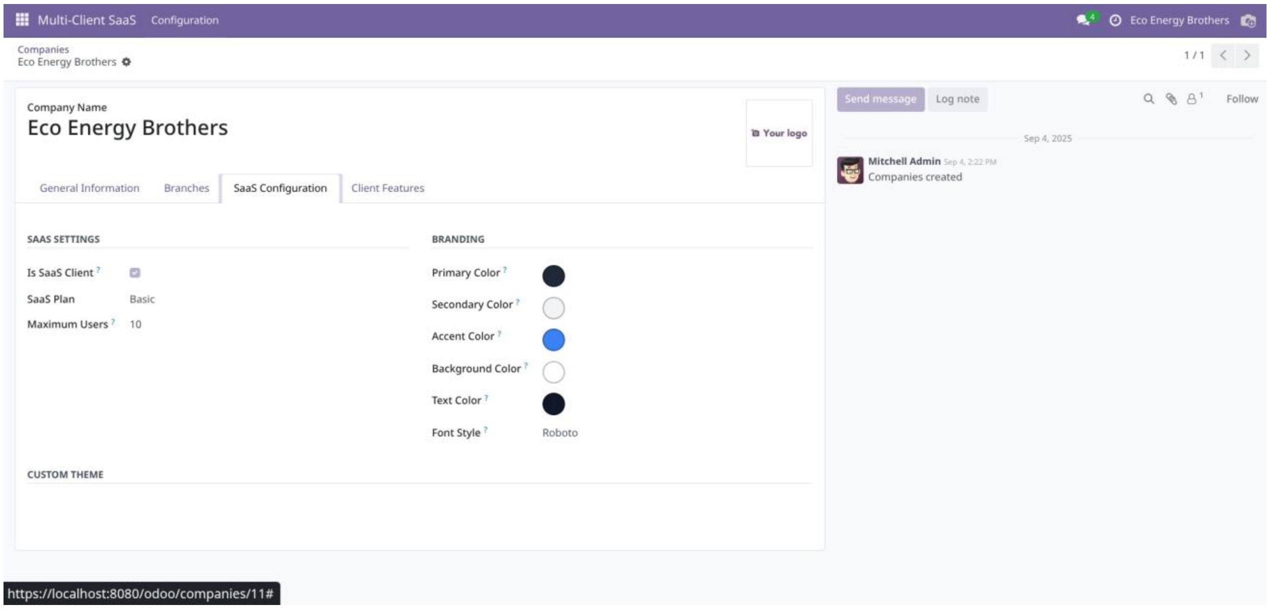

CRM 216 - Complete Business Suite

Complete business suite installer for CRM 216

Multi-Client SaaS Platform

Multi-tenant SaaS platform with client-specific branding and features

Figure 10: Overview of Odoo modules, illustrating the modular architecture of the platform.



Multi-Client SaaS Configuration

Companies
Eco Energy Brothers

Company Name
Eco Energy Brothers

General Information | Branches | SaaS Configuration | Client Features

SAAS SETTINGS

- Is SaaS Client: ☑
- SaaS Plan: Basic
- Maximum Users: 10

BRANDING

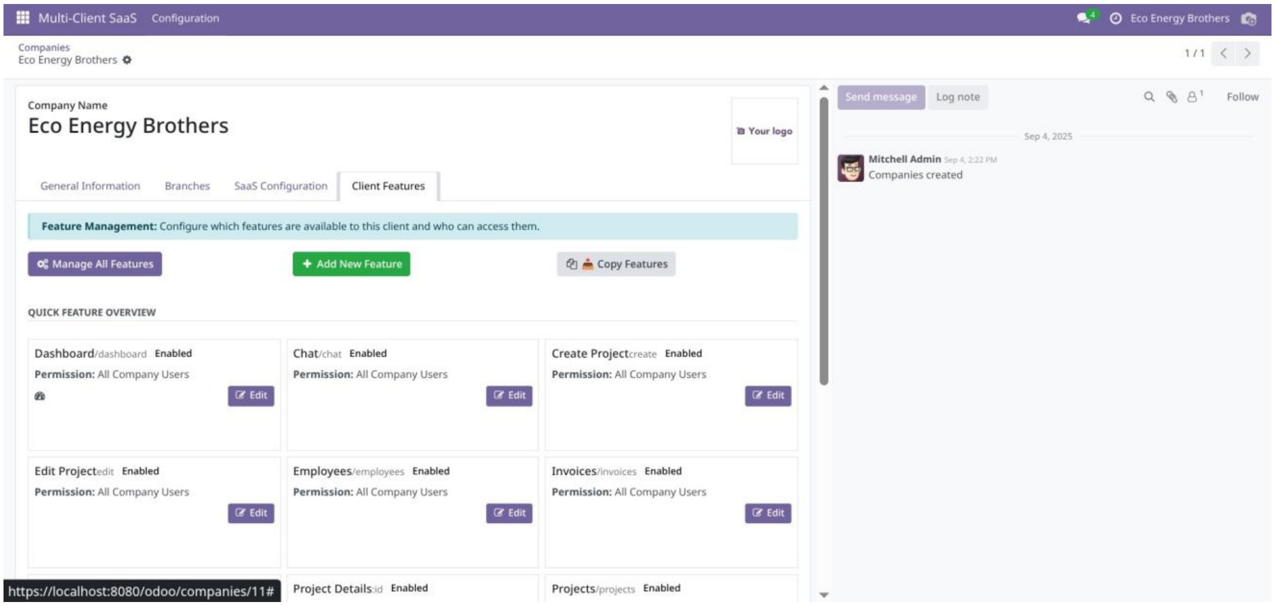
- Primary Color: ●
- Secondary Color: ○
- Accent Color: ●
- Background Color: ○
- Text Color: ●
- Font Style: Roboto

CUSTOM THEME

Mitchell Admin Sep 4, 2:22 PM
Companies created

https://localhost:8080/odoo/companies/11#

Figure 11: Details of Odoo SaaS configuration, showing multi-tenant setup options.



Multi-Client SaaS Configuration

Companies
Eco Energy Brothers

Company Name
Eco Energy Brothers

General Information | Branches | SaaS Configuration | Client Features

Feature Management: Configure which features are available to this client and who can access them.

Manage All Features | Add New Feature | Copy Features

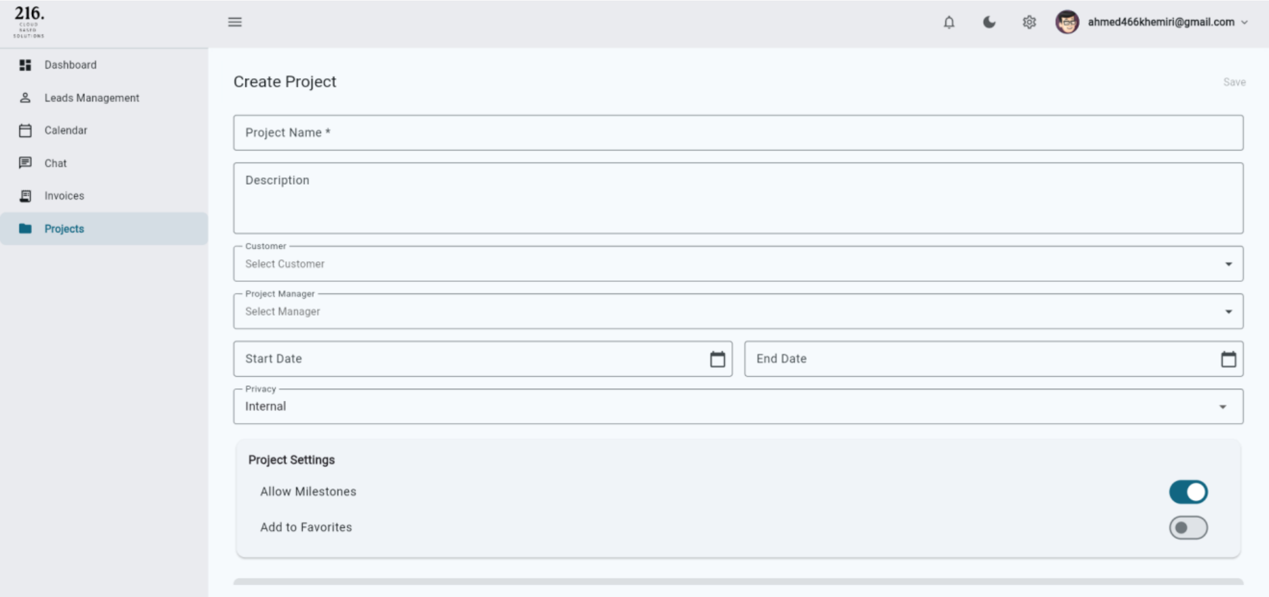
QUICK FEATURE OVERVIEW

- Dashboard/dashboard Enabled — **Permission:** All Company Users
- Chat/chat Enabled — **Permission:** All Company Users
- Create Projectcreate Enabled — **Permission:** All Company Users
- Edit Projectedit Enabled — **Permission:** All Company Users
- Employees/employees Enabled — **Permission:** All Company Users
- Invoices/invoices Enabled — **Permission:** All Company Users
- Project Details:id Enabled
- Projects/projects Enabled

Mitchell Admin Sep 4, 2:22 PM
Companies created

https://localhost:8080/odoo/companies/11#

Figure 12: Odoo SaaS features page highlighting cloud-based functionalities.



Create Project

Project Name *

Description

Customer: Select Customer

Project Manager: Select Manager

Start Date

End Date

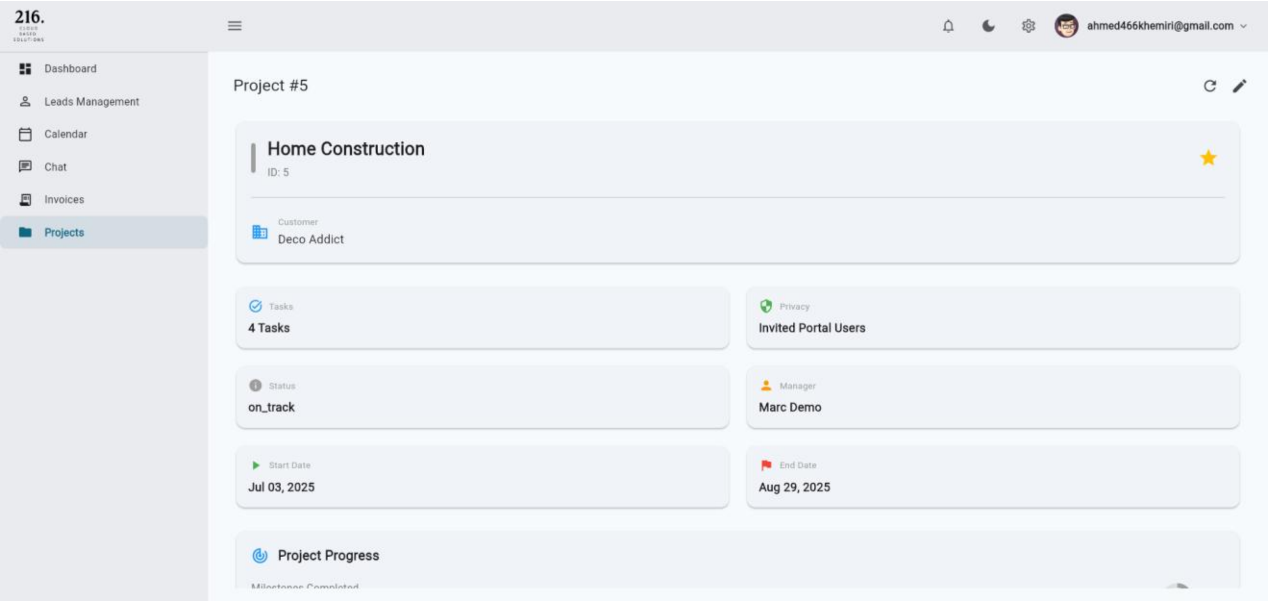
Privacy: Internal

Project Settings

Allow Milestones ●

Add to Favorites ○

Figure 13: Project creation form where users can define new projects and assign resources.



Project #5

Home Construction

ID: 5

Customer: Deco Addict

Tasks: 4 Tasks

Privacy: Invited Portal Users

Status: on_track

Manager: Marc Demo

Start Date: Jul 03, 2025

End Date: Aug 29, 2025

Project Progress

Milestones Completed

Figure 14: Project details interface showing tasks, deadlines, and project structure.

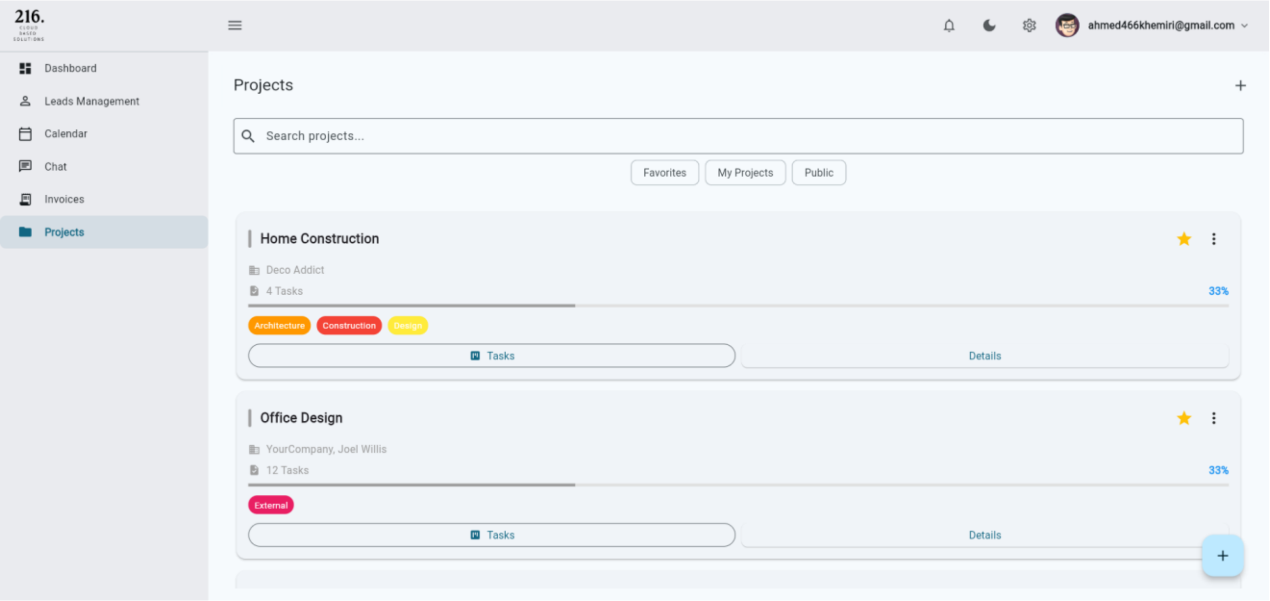


Figure 15: Project list view presenting all projects managed within the platform.

5 LITERATURE REVIEW

This section situates my internship project within existing research and best practices in the fields of SaaS systems, web application development, and CRM software design. The review draws on theoretical frameworks and applied studies to contextualize the technical decisions made during the internship and to critically reflect on their strengths and limitations.

5.1 SaaS Architecture and Multi-Tenancy

The idea of Software as a Service (SaaS), in which software is provided as a subscription-based service as opposed to a one-time installation, forms the basis of the 216.CBS initiative. Multi-tenancy is a crucial difficulty in SaaS system design, according to research: in order to be efficient, applications must separate client data while using the same infrastructure. According to studies, permission-based access control and database-level separation are popular methods for ensuring tenant isolation; nevertheless, they also carry the danger of performance bottlenecks in situations with high concurrency. These trade-offs were evident in my work on Odoo modules, where I had to strike a balance between responsiveness and rigorous isolation during subscription validation.

5.2 Web Application Development Frameworks

The choice of Flutter for Web aligns with recent literature advocating for cross-platform frameworks to reduce development overhead. Flutter's reactive UI model and strong support for code reuse make it a viable alternative to traditional web frameworks. It was the best choice for our scenario, since we weren't so sure about how we would scale in the future and which devices we will be adding support to. However, literature also notes performance limitations for highly complex web applications compared to native frameworks. My experience supported this view: Flutter allowed rapid development of a consistent CRM interface, it was worth the trade-off since expanding in the future will be much faster since Flutter is cross-platform.

5.3 State Management and Routing

Modern web apps require predictable state management to handle dynamic data and user interactions. Literature on application architecture emphasizes that predictable state management frameworks (e.g., Redux, Provider, Riverpod) simplify debugging and enhance maintain- ability in complex apps. Similarly, permission-aware routing is described as a crucial practice in multi-user systems to ensure that navigation aligns with user roles and access rights. My implementation of Riverpod providers and GoRouter followed these established principles, validating their relevance in production-grade SaaS applications.

5.4 Data Validation and Security Models

Academic and industry research stresses the importance of server side validation to protect SaaS platforms from threats such as session hijacking, privilege escalation, and data corruption. Client-side validation is useful for user experience but insufficient for security. My project mirrored these best practices by relying on server-side subscription checks, complemented by client-side sanitization of over 200 fields. This approach aligns with literature advocating for defense in depth, where multiple layers of validation reduce the likelihood of successful attacks.

5.5 CRM Systems and Usability

The role of Customer Relationship Management (CRM) software in supporting small and medium enterprises has been widely studied. Research highlights that successful CRM platforms must combine robust backend functionality with intuitive user interfaces. Dashboards and reporting tools, in particular, are noted as key features for improving decision-making. Additionally, I've decided to develop the CRM for web platforms / browsers to support a broader range of devices, since some of the clients that we were targeting are restaurants or coffee shops, most of which use Android Tablets instead of computers. Targeting web browsers is more efficient than targeting Android and Windows, especially at the start of the product's life.

6 METHODOLOGY

I adopted a hands-on approach during my internship at Alwen Co., which allowed me to apply both project management and software engineering methods to the development of the 216.CBS CRM system. Because this was a subscription-based web application that integrated a customized Odoo backend with a Flutter frontend, my methodology combined iterative development practices with structured testing and documentation.

Development Methodology

The team followed Agile practices inspired by Scrum to organize our work. Development was broken down into short sprints, each focused on incremental deliverables such as subscription validation, deployment automation, or routing integration. At the beginning of each sprint, tasks were discussed and prioritized, and daily stand-up meetings ensured progress was tracked and blockers were addressed quickly. This iterative approach was essential, as requirements often evolved in response to client feedback or technical constraints. Agile gave us the flexibility to adapt while maintaining alignment with the overall project goals of delivering a scalable SaaS CRM platform.

Software Development Techniques

On the backend, I worked primarily with Odoo v18, extending its framework through custom modules. The Multi Client SaaS module leveraged Odoo’s security constructs, including groups, record rules, and model access rights, to enforce tenant-level isolation. The 216.CRM Suite module automated installation tasks by chaining module dependencies and initializing system parameters with XML data files and server actions. My approach emphasized modularity and reusability so that new tenants could be onboarded with minimal manual intervention.

On the frontend, I used Flutter for Web to implement a responsive, cross-platform interface. I structured the app with Riverpod for state management and GoRouter for navigation, ensuring a clear separation between authentication routes, feature routes, and core application flows. To handle server communication, I implemented the Odoo Service layer using Dio with interceptors, which provided session persistence, error parsing, and retry/backoff strategies. By centralizing communication through this layer, I ensured consistent request handling and simplified debugging.

Data Management and Integration

Data management was centered on Odoo’s ORM, which stores client, subscription, and CRM records in a PostgreSQL database. I designed data models for tenant subscriptions and feature registries, allowing efficient toggling of functionality per client. On the frontend, I created a field registry that maps Odoo field definitions to type-safe Dart models. This registry supports constant- time lookups and applies sanitizers to over 95% of the system’s 200+ fields, reducing the risk of invalid or unsafe inputs reaching the backend. The integration relied on Odoo’s JSON-RPC API, which was extended by our custom endpoints for subscription validation and feature availability checks.

User Testing and Feedback

Testing was a continuous part of development. I implemented **unit tests** for every module I’ve implemented, like Employees or Projects, making sure that permission-based routing, edits, and access worked as intended. According to internal project documentation, all eight-unit tests passed successfully, validating field lookups, registry initialization (<10 ms for three models), and permission-based routing logic. Structured logging with contextual metadata was used to monitor API calls and response times, which helped identify performance bottlenecks under load. Informal feedback loops within the development team also guided improvements, particularly in the user-facing modules such as Sales and Projects.

Conclusion

In summary, my methodology combined Agile iteration, modular development, and rigorous testing. On the backend, I emphasized tenant isolation and automation to ensure scalability. On the frontend, I focused on secure session management, permission-aware routing, and reliable data handling. This approach allowed me not only to contribute concrete features to the 216.CBS CRM platform but also to gain deeper insights into best practices for SaaS development. The methodology highlighted the value of collaboration, adaptability, and systematic validation, all of which were crucial in shaping the success of the project.

7 RESULTS AND FINDINGS

During my internship at Alwen Co., I concentrated on two main aspects of the 216.CBS project: building custom Odoo modules for multi-tenant SaaS deployment and developing the Flutter-based web frontend. These contributions gave me practical experience across backend design, secure API communication, and permission-aware user interface development. The most significant results are presented in the following subsections, with figures highlighting the workflows and application interfaces I implemented.

Backend Development

Multi-Client SaaS Module

The Multi-Client SaaS module manages subscription-based feature access. It links client tiers to Odoo permissions, ensuring that users can only access functionality defined by their subscription. The data model connects tenants, features, and permissions, while record rules restrict data visibility at the database level. This enforces strict data isolation and prevents unauthorized access while supporting multiple tenants on a single platform.

CRM Suite Deployment Module

The 216.CRM Suite module automates the installation and configuration of the CRM stack. It ensures dependencies are installed in the correct order, initializes company parameters, and provisions production-ready client instances. Automating this sequence reduced setup time and standardized deployment across multiple clients.

Frontend Web Application

Functional Modules

I contributed to several modules in the CRM system, each providing distinct functionality:
Employees: Users can add new employees, view detailed personal and work-related information, and browse the complete employee list. This ensures HR managers can efficiently manage staff records (Figures 1–3).
Invoices: The module allows creating invoices, viewing detailed invoice records, and browsing all invoices in the system. Users can track invoice status and line items, facilitating accurate financial management (Figures 4–7).

Odoo Features: This module enables creating and replicating features across the platform. Users can quickly duplicate existing workflows or add new functionalities, improving system flexibility and maintainability (Figures 8–9).

Odoo Modules Overview: Provides a high-level view of the modular architecture of the platform, making it easy to manage installed modules and dependencies (Figure 10).

Odoo SaaS: Users can configure multi-tenant options and access cloud-based features, supporting scalable SaaS deployments (Figures 11–12).

Projects: Enables creation of new projects, managing project details such as tasks and deadlines, and browsing all projects within the system. This facilitates efficient planning and resource allocation (Figures 13–15).

Performance and Testing

Performance metrics demonstrated the speed and dependability of typical activities across different modules. Response times for user interactions, data retrieval, and registry initialization are constant and appropriate for small to medium datasets. Error monitoring and logging verified consistent behavior under numerous concurrent interactions, guaranteeing the system is ready for daily use and production.

8 RECOMMENDATIONS

Based on the results of my internship, I recommend several improvements to strengthen the scalability, reliability, and user experience of the 216.CBS CRM platform. First, enhancing the deployment automation in the 216.CRM Suite module with stronger rollback and idempotency safeguards would make multi-tenant provisioning more reliable and reduce the risk of partial installations. Second, adopting advanced caching strategies in the web application could improve performance when handling large datasets, especially in modules like Sales and Invoices. Third, introducing real-time synchronization through WebSockets or a message broker would allow faster updates than the current polling approach, though this would require deeper integration with Odoo. Finally, implementing schema validation between Odoo models and the field registry would further increase data integrity, ensuring that new features can be added without compromising stability.

9 CONCLUSION

By improving backend Odoo modules and putting in place a Flutter-based web interface, I helped Alwen Co. construct a multi-tenant SaaS CRM during my internship. In addition to giving me excellent chances to practice collaboration, problem-solving, and actual software deployment, this experience improved my abilities in scalable system design, secure API integration, and cross-platform programming. All things considered, the internship provided a thorough understanding of creating reliable, enterprise-ready applications.

REFERENCES

[1] Odoo S.A., *Odoo Documentation (v18)*, [Online].

Available: https://www.odoo.com/documentation. [Accessed: Sept. 25, 2025].

[2] Google LLC, *Flutter Documentation*, [Online].

Available: https://docs.flutter.dev/. [Accessed: Sept. 25, 2025].

[3] Riverpod.dev, *Riverpod State Management Documentation*, [Online].

Available: https://riverpod.dev/. [Accessed: Sept. 25, 2025].

[4] GoRouter.dev, *GoRouter: Declarative Routing for Flutter*, [Online].

Available: https://pub.dev/packages/go_router. [Accessed: Sept. 25, 2025].

[5] D. Crockford, *The Application/JSON-RPC Specification*, 2005. [Online].

Available: https://www.jsonrpc.org/specification. [Accessed: Sept. 25, 2025].

[6] K. Beck and C. Andres, *Extreme Programming Explained: Embrace Change*, 2nd ed., Boston, MA: Addison-Wesley, 2004.

[7] K. Schwaber and J. Sutherland, *The Scrum Guide*, Scrum.org, 2020. [Online].

Available: https://scrumguides.org/. [Accessed: Sept. 25, 2025].