🧩 Project Case Study: Custom CRM SaaS Platform (Legacy to Laravel + React Migration)

📘 Project Overview

Designed and developed a full-featured, modular **CRM SaaS platform** tailored for small to medium-sized businesses. The platform empowers teams to manage contacts, leads, tasks, quotes, invoices, and communication — all within a secure, scalable multi-user environment.

Initially built using **CodeIgniter + jQuery**, I migrated the entire system to a modern **Laravel + React architecture**, introducing REST APIs, component-based UI, and advanced role-based access control (RBAC).

🎯 Objectives & Business Goals

- Migrate legacy monolithic system to modern service-based architecture

- Enable seamless contact and lead management with communication tracking

- Automate recurring billing, invoicing, and quote generation

- Introduce multi-role user access with fine-grained permission control

- Improve system performance and API extensibility for future mobile integrations

---

🚧 Challenges Faced

- Outdated CodeIgniter structure lacked modularity and API support

- jQuery-based frontend caused slow rendering and poor UX

- No centralized RBAC or user access control

- Incomplete communication tracking (emails, SMS)

- Lack of proper DevOps and deployment automation

🛠️ My Role & Technology Stack

- 🔹 **My Role**

Full-Stack Engineering Manager & SaaS Architect
Owned the full software lifecycle: architecture, API design, backend services, frontend development, DevOps, and third-party integrations.

- ◆ **Backend Stack**

  - **Laravel 10+**, RESTful APIs, Service Providers

  - **MySQL**, Redis, Task Queues

  - **Twilio** (SMS), **AWS SES** (email), **CloudWatch** (logs)

- ◆ **Frontend Stack**

  - **React.js**, **Tailwind CSS**, Formik, Axios

  - Responsive layout, reusable components, SPA patterns

- ◆ **DevOps & Tools**

  - Docker, GitHub Actions, AWS EC2, Laravel Forge, CloudWatch

  - Postman collection + OpenAPI documentation for frontend/mobile teams

---

⚙️ Solution & Architecture

✅ Authentication & User Management

- Token-based auth using Laravel Passport

- Forgot/reset password, OTP verification via email/SMS

- User profile edit, avatar, and contact preferences

✅ Modular Service-Based APIs

- Built **70+ endpoints** organized by service: Contact, Lead, Quote, Invoice, Task, Pipeline, Settings, Dashboard

- Followed **RESTful conventions**, integrated activity logs and audit trails

- Enabled role-specific API access using policy gates and middleware

✅ CRM Core Modules

- **Contact Management**: CRUD, custom fields, address history, notes, phone/email logs

- **Lead Management**: Lead tracking, pipelines, stages, tags

- **Quote & Invoice Builder**: PDF generation, version control, payment logs

- **Task & Schedule System**: Time tracking, calendar views, server-time sync

- **Dashboard Reports**: Activities, task stats, user performance

✅ **Communication Engine**

- **SMS integration with Twilio**

- **Email templates via AWS SES**

- Bulk messaging, email history, and bounce handling

✅ **Custom Field & Template Builder**

- Dynamic form builder for leads/contacts

- Role-based field visibility & edit permissions

- Email template system with variable support and preview

✅ **DevOps & Logging**

- Dockerized setup with staging and production parity

- GitHub Actions CI/CD pipeline with auto-deploy

- **CloudWatch logging** for API monitoring and error tracking

---

🧪 Testing & QA

- Manual test cases documented in Notion

- Used Postman for API testing with dynamic variables

- Error handling and fallback scenarios for SMS/email failures

- Activity logs to trace user actions and API consumption

📈 Results & Business Impact

✅ Migrated legacy CRM to modern Laravel 10 + React SPA without downtime

✅ Reduced API response time by **60%** via query optimizations and caching

✅ Delivered full CRM MVP in **under 40 days**

✅ Onboarded 500+ users within 3 months

✅ Enabled SMS & email communication tracking with delivery logs

✅ Improved task efficiency for users with calendar filters and time tracking

✅ Designed for mobile app integration via REST API with tenant-aware headers

💡 What I Learned

- **Modernizing Legacy Systems:** Successfully transformed a monolithic CodeIgniter app into a modular, scalable Laravel + React platform — deepening my understanding of service-based architectures and API design.

- **Advanced Role-Based Access Control:** Implemented fine-grained RBAC policies, improving security and user management in a multi-tenant SaaS environment.

- **Optimizing Performance:** Learned how to reduce API response times by 60% through efficient query optimization and caching strategies.

- **Communication Integrations:** Mastered integrating external services like Twilio and AWS SES to build a robust communication engine with SMS and email tracking.

- **DevOps & CI/CD:** Gained hands-on experience with Docker, GitHub Actions, and Laravel Forge for smooth, zero-downtime deployments.

- **User-Centric Design:** Improved UI/UX using React component patterns and Tailwind CSS, focusing on responsiveness and reusability.

- **Testing & QA:** Strengthened API testing with Postman, error handling, and detailed activity logs for audit and debugging.

📌 Key Takeaways

- Service-based architecture in Laravel makes the system highly maintainable and scalable

- React's component-based structure made the UI modular and lightning fast

- A robust communication engine (Twilio + SES) dramatically improved engagement rates

- GitHub Actions + Laravel Forge simplified CI/CD with zero-downtime deployments

---

🔗 Access & Demo

- 🔍 **Postman Collection –** Available privately upon request or NDA

- 🚀 **Live Demo / Staging –** Secure environment access available during interviews or demos

- 📂 **GitHub Repository –** Code access can be shared selectively (sanitized version available on request)

🔙 Conclusion

This project not only showcases my ability to architect and deliver complex, scalable SaaS applications but also highlights my commitment to driving business impact through modern technology and best practices. Migrating a legacy system to a robust Laravel + React platform has enabled enhanced user experience, improved performance, and streamlined operations for the client.

I am passionate about leveraging my full-stack expertise to contribute effectively to your projects and tech stack—whether it's building scalable web applications, optimizing backend APIs, or designing seamless user interfaces. I welcome opportunities to collaborate, innovate, and add value to your team or organization.

Feel free to reach out if you're looking for a dedicated engineer ready to deliver impactful solutions and accelerate your product's success. I'm excited to assist with your next challenge!